



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE GRADO

DISEÑO DE UN SISTEMA DE MONITORIZACIÓN DE DISPOSITIVOS MÓVILES

Autor: Antonio Becerra López

Tutor: Luis Miguel Sánchez García

Colmenarejo, Septiembre de 2012

Título: Diseño de un sistema de monitorización de dispositivos móviles

Autor: Antonio Becerra López

Director:

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ____ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Resumen

AndroidMonitor es un sistema cuya finalidad consiste en monitorizar distribuidamente dispositivos móviles con el sistema operativo Android.

Los teléfonos móviles en la actualidad son un elemento indispensable más de las personas junto con las llaves de casa y la cartera con los documentos personales. En el año 2011, la población total de la Tierra superaba la asombrosa cantidad de siete mil millones de personas. En ese mismo año, se calculaba que el número de teléfonos móviles en el mundo era superior a los 5.600 millones. Los dispositivos móviles cuentan con gran variedad de sensores: acelerómetro, sensor de luz, sensor de temperatura, barómetro, sensor de gravedad, sensor de presión; Sensores que recopilan información de su entorno que puede ser utilizada para comprar la situación del terminal.

El objetivo de este proyecto es el análisis, el diseño y el desarrollo de un sistema para monitorizar remotamente los valores de los sensores de dispositivos móviles. Los usuarios de la aplicación podrán monitorizar sus dispositivos registrado desde terminales remotos para controlar así el estado de sus terminales móviles.

Palabras clave: AndroidMonitor, Android, Monitorización.

Índice de contenido

Índice de ilustraciones	6
Índice de Tablas	10
1 Introducción	14
1.1 Motivación	14
1.2 Objetivos	15
1.3 Estructura del documento	16
1.4 Siglas y Acrónimos	16
1.5 Glosario de términos.....	18
2 Estado de la cuestión	19
2.1 Sistemas operativos móviles.....	19
2.1.1 Android	19
2.1.2 iOS.....	28
2.1.3 Windows Phone.....	30
2.1.4 Blackberry OS.....	32
2.1.5 Symbian	33
2.1.6 WebOS	35
2.2 Servicios web	38
2.2.1 REST	38
2.2.2 SOAP	41
2.2.3 Comparativa	42
2.3 Aplicaciones de monitorización	45
2.3.1 SCADA	45
3 Marco regulador.....	47
4 Análisis.....	48

4.1	Metodología.....	48
4.2	Análisis del proyecto.....	49
4.2.1	Modelo de casos de uso	49
4.2.2	Análisis de requisitos	58
4.2.3	Análisis de bases de datos	75
5	Diseño.....	81
5.1	Lenguaje de programación	81
5.1.1	Servidor web.....	81
5.1.2	Aplicación móvil.....	82
5.2	Base de datos.....	82
5.2.1	Aplicación móvil.....	82
5.2.2	Servidor web.....	85
5.3	Comunicaciones	90
5.3.1	Registro.....	91
5.3.2	Cambio de periodo	92
5.3.3	Cambio de contraseña.....	93
5.3.4	Monitorización.....	93
5.4	Aplicación Servicio	94
5.5	Interfaces de usuario	96
5.5.1	Aplicación móvil.....	96
5.5.2	Servidor web.....	103
6	Desarrollo	109
6.1	Servidor web	109
6.1.1	Servicio Web	109
6.1.2	Aplicación Web	114
6.2	Aplicación móvil	126

6.2.1	Cliente Web	126
6.2.2	Aplicación	133
7	Pruebas de evaluación	142
7.1	Tiempo de ejecución.....	142
7.1.1	Android Service.....	142
7.1.2	Servicio Web	144
7.1.3	Comunicación	145
7.1.4	Conclusiones de los análisis.....	146
8	Conclusiones.....	147
8.1	Conclusiones obtenidas	147
8.2	Trabajos futuros.....	147
8.3	Presupuesto del proyecto.....	149
8.4	Planificación	150
9	Bibliografía	152
10	Anexos	155
10.1	Anexo I: Manual de instalación.....	155
10.1.1	Aplicación web.....	155
10.1.2	Aplicación móvil.....	162
10.2	Anexo II: Manual de usuario	167
10.2.1	Aplicación móvil.....	167
10.2.2	Aplicación Web	174
10.3	Anexo II: Contenido del disco	179

Índice de ilustraciones

Ilustración 1.1: Dispositivos móviles	14
Ilustración 2.1: Android logo	19
Ilustración 2.2: Nexus One con Android 2.1	20
Ilustración 2.3: Interfaz de Ice Cream Sandwich	20
Ilustración 2.4: Distribución actual de versiones	21
Ilustración 2.5: Galaxy Nexus y Nexus 7 con Android 4.1	23
Ilustración 2.6: Arquitectura Android.....	24
Ilustración 2.7: Ciclo de vida de un Activity.....	26
Ilustración 2.8: iOS logo.....	28
Ilustración 2.9: iPhone 4S con iOS 5	28
Ilustración 2.10: Interfaz de iOS	29
Ilustración 2.11: Arquitectura iOS.	30
Ilustración 2.12: Windows Phone logo.....	30
Ilustración 2.13: Nokia Lumia 900 con Windows Phone 7	30
Ilustración 2.14: Arquitectura Windows Phone	32
Ilustración 2.15: Blackberry logo	32
Ilustración 2.16: Blackberry Bold 9790 con Blackberry OS 7	33
Ilustración 2.17: Symbian logo	33
Ilustración 2.18: Nokia N8 con Symbian^3.....	34
Ilustración 2.19: Arquitectura Symbian.....	35
Ilustración 2.20: HP WebOS logo.....	35
Ilustración 2.21: HP Touchpad con WebOs 3	35
Ilustración 2.22: Interfaz Palm WebOS.....	36
Ilustración 2.23: Arquitectura WebOS simplificada	37
Ilustración 2.24: Arquitectura WebOS	37
Ilustración 2.25: Servicio Web REST	38
Ilustración 2.26: Axis logo.....	39
Ilustración 2.27: Axis2 logo.....	40
Ilustración 2.28: Restlet logo	40
Ilustración 2.29: Estructura SOAP.....	42

Ilustración 2.30: Servicios Web	44
Ilustración 2.31: Un sistema SCADA típico	46
Ilustración 4.1: eXtreme Programming	48
Ilustración 4.2: Modelo de casos de uso	57
Ilustración 4.3: Modelo entidad-relación de la base de datos de la aplicación móvil	75
Ilustración 4.4: Modelo entidad-relación de la base de datos del servidor	77
Ilustración 5.1: Diagrama de la base de datos de la aplicación móvil	83
Ilustración 5.2: Diagrama de la base de datos del servidor	85
Ilustración 5.3: Comunicación del sistema	90
Ilustración 5.4: Diagrama de secuencia de las comunicaciones	91
Ilustración 5.5: Diagrama de secuencia del servicio de la Aplicación móvil	93
Ilustración 5.6: Diagrama de flujo del servicio de la aplicación móvil	95
Ilustración 5.7: Navegación entre pantallas de la aplicación móvil	96
Ilustración 5.8: Interfaz de registro de la aplicación móvil	97
Ilustración 5.9: Interfaz de usuario del login de la aplicación móvil	98
Ilustración 5.10: Interfaz del menú de la aplicación móvil	99
Ilustración 5.11: Interfaz del perfil de la aplicación móvil	100
Ilustración 5.12: Interfaz de la monitorización de la aplicación móvil	101
Ilustración 5.13: Interfaz de la configuración de la aplicación móvil	102
Ilustración 5.14: Navegación entre pantallas de la página Web	103
Ilustración 5.15: Interfaz del índice de la página Web	104
Ilustración 5.16: Interfaz de la página descargar app de la página Web	105
Ilustración 5.17: Interfaz de la página panel de la página web	106
Ilustración 5.18: Interfaz de la página detalles de la página Web	107
Ilustración 5.19: Interfaz de la página histórico de la página Web	108
Ilustración 6.1: Captura de pantalla de la página índice de la Web	116
Ilustración 6.2: Captura de pantalla de la página descarga de la Web	117
Ilustración 6.3: Captura de pantalla de la página panel de la Web	122
Ilustración 6.4: Captura de pantalla de la página detalles de la Web	124
Ilustración 6.5: Captura de pantalla de la página histórico de la Web	125
Ilustración 6.6: Navegación de pantallas en la página Web	126
Ilustración 6.7: Captura de pantalla de la interfaz Registro de la aplicación	128

Ilustración 6.8: Captura de pantalla de la interfaz identificación de la aplicación	134
Ilustración 6.9: Captura de pantalla de la interfaz menú de la aplicación	135
Ilustración 6.10: Captura de pantalla de la interfaz perfil de la aplicación.....	136
Ilustración 6.11: Captura de pantalla de la interfaz monitor de la aplicación	138
Ilustración 6.12: Captura de pantalla de la interfaz configuración de la aplicación	140
Ilustración 6.13: Navegación de pantallas en la aplicación móvil.....	140
Ilustración 7.1: Pruebas de comunicación.....	145
Ilustración 8.1: Planificación Gantt.....	151
Ilustración 10.1: Manual de instalación: archivo interfaces	156
Ilustración 10.2: Manual de instalación: archivo interfaces modificado	156
Ilustración 10.3: Manual de instalación: LAMP	157
Ilustración 10.4: Manual de instalación: Servidor Web	157
Ilustración 10.5: Manual de instalación: PhpMyadmin Instalación	158
Ilustración 10.6: Manual de instalación: PhpMyAdmin instalación 2	158
Ilustración 10.7: Manual de instalación: PhpMyAdmin login	159
Ilustración 10.8: Manual de instalación: crear la base de datos	159
Ilustración 10.9: Manual de instalación: importar código	160
Ilustración 10.10: Manual de instalación: Subida de la Web	161
Ilustración 10.11: Manual de instalación: Android SDK manager	162
Ilustración 10.12: Manual de instalación: instalar nuevo software en Eclipse	163
Ilustración 10.13: Manual de instalación: instalar el plugin.....	163
Ilustración 10.14: Manual de instalación: selección de paquetes	164
Ilustración 10.15: Manual de instalación: configurar SDK	164
Ilustración 10.16: Manual de Instalación: importar proyecto 1	165
Ilustración 10.17: Manual de instalación: importar proyecto 2.....	165
Ilustración 10.18: Manual de usuario: Menú Android 2.3.6	167
Ilustración 10.19: Manual de usuario: Registro en la aplicación	168
Ilustración 10.20: Manual de usuario: dato mal introducido.....	168
Ilustración 10.21: Manual de usuario: Identificación de usuarios	169
Ilustración 10.22: Manual de usuario: Menú principal	170
Ilustración 10.23: Manual de usuario: Pagina perfil de usuario	170
Ilustración 10.24: Manual de usuario: Pantalla monitorizar	171

Ilustración 10.25: Manual de usuario: Pantalla opciones	172
Ilustración 10.26: Manual de usuario: Identificación en la página Web.....	174
Ilustración 10.27: Manual de usuario: Página panel de la Web.....	175
Ilustración 10.28: Manual de usuario: Descargar aplicación	175
Ilustración 10.29: Selector de idiomas	176
Ilustración 10.30: Manual de usuario: Página historico.	176
Ilustración 10.31: Manual de usuario: Página de monitorización detallada.....	177

Índice de Tablas

Tabla 1.1: Siglas y acrónimos.....	17
Tabla 2.1: Distribución actual de versiones Android.....	22
Tabla 2.2: REST vs SOAP	43
Tabla 4.1: Caso de uso 1: Descargar aplicación móvil	50
Tabla 4.2: Caso de uso 2: Registrarse en la aplicación	51
Tabla 4.3: Caso de uso 3: Identificarse en la aplicación	51
Tabla 4.4: Caso de uso 4: Ver perfil	52
Tabla 4.5: Caso de uso 5: Activar/desactivar monitorización	52
Tabla 4.6: Caso de uso 6: Cambiar periodo.....	53
Tabla 4.7: Caso de uso 7: Monitorizar desde terminal.....	53
Tabla 4.8: Caso de uso 8: Identificarse en el servidor web	54
Tabla 4.9: Caso de uso 9: Monitorización distribuida	54
Tabla 4.10: Caso de uso 10: Ver detalles de monitorización	55
Tabla 4.11: Caso de uso 11: Consultar histórico	55
Tabla 4.12: Caso de uso 12: Seleccionar idioma	56
Tabla 4.13: Caso de uso 13: Cambiar contraseña	56
Tabla 4.14: Requisito de usuario 1: Registro	58
Tabla 4.15: Requisito de usuario 2: Identificación en la aplicación móvil	59
Tabla 4.16: Requisito de usuario 3: Identificación en la aplicación web	59
Tabla 4.17: Requisito de usuario 4: Descarga aplicación	59
Tabla 4.18: Requisito de usuario 5: Ver datos de usuario.....	59
Tabla 4.19: Requisito de usuario 6: Cambiar estado de la monitorización.....	60
Tabla 4.20: Requisito de usuario 7: Cambiar periodo de la monitorización	60
Tabla 4.21: Requisito de usuario 8: Monitorizar dispositivo.....	60
Tabla 4.22: Requisito de usuario 9: Monitorizar distribuida del dispositivo.....	60
Tabla 4.23: Requisito de usuario 10: Monitorización detallada.....	61
Tabla 4.24: Requisito de usuario 11: Ver histórico.....	61
Tabla 4.25: Requisito de usuario 12: Seleccionar idioma.....	61
Tabla 4.26: Requisito de usuario 13: Cambiar contraseña.....	62
Tabla 4.27: Requisito de restricción 1: Registro en aplicación.....	62

Tabla 4.28: Requisito de restricción 2: Identificación en el servidor	62
Tabla 4.29: Requisito de restricción 3: Registro	62
Tabla 4.30: Requisito de restricción 4: Idiomas.....	63
Tabla 4.31: Requisito de restricción 5: Cambio de contraseña	63
Tabla 4.32: Trazabilidad: casos de uso – requisitos de usuario	64
Tabla 4.33: Requisito de software 1: Registro.....	65
Tabla 4.34: Requisito de software 2: identificación aplicación	65
Tabla 4.35: Requisito de software 3: Identificación web	65
Tabla 4.36: Requisito de software 4: descarga aplicación	65
Tabla 4.37: Requisito de software 5: Datos de perfil	66
Tabla 4.38: Requisito de software 6: monitorización.....	66
Tabla 4.39: Requisito de software 7: monitorización sin conexión	66
Tabla 4.40: Requisito de software 8: Modificación de la monitorización	66
Tabla 4.41: Requisito de software 9: Introducir periodo	67
Tabla 4.42: Requisito de software 10: cambiar periodo	67
Tabla 4.43: Requisito de software 11: Monitorización desde dispositivo	67
Tabla 4.44: Requisito de software 12: Monitorización distribuida	67
Tabla 4.45: Requisito de software 13: Monitorización detallada	68
Tabla 4.46: Requisito de software 14: Histórico	68
Tabla 4.47: Requisito de software 15: Autenticación	68
Tabla 4.48: Requisito de software 16: Autenticación web	69
Tabla 4.49: Requisito de software 17: Dispositivos asociados.....	69
Tabla 4.50: Requisito de software 18: Cambiar idioma	69
Tabla 4.51: Requisito de software 19: Idiomas	69
Tabla 4.52: Requisito de software 20: Cambiar contraseña	70
Tabla 4.53: Requisito de software 21: Cambio de contraseña.....	70
Tabla 4.54: Requisito de software 22: Lenguajes de programación	70
Tabla 4.55: Requisito de software 23: Lenguaje aplicación	70
Tabla 4.56: Requisito de software 24: Versión Sistema operativo	71
Tabla 4.57: Requisito de software 25: Contraseñas	71
Tabla 4.58: Requisito de software 26: Conexión segura	71
Tabla 4.59: Tabla de correspondencia de requisitos.....	72

Tabla 4.60: Matriz de consistencia	74
Tabla 4.61: Entidad Usuario móvil.....	76
Tabla 4.62: Entidad Configuración	76
Tabla 4.63: Entidad Sensores.....	76
Tabla 4.64: Entidad Mensajes pendientes	76
Tabla 4.65: Entidad Usuario servidor	77
Tabla 4.66: Entidad Dispositivos.....	78
Tabla 4.67: Entidad Giroscopio.....	78
Tabla 4.68: Entidad Velocidad de rotación.....	78
Tabla 4.69: Entidad Presión	78
Tabla 4.70: Entidad Aceleración lineal	79
Tabla 4.71: Entidad Proximidad.....	79
Tabla 4.72: Entidad Acelerómetro.....	79
Tabla 4.73: Entidad Campo magnético.....	79
Tabla 4.74: Entidad Temperatura.....	80
Tabla 4.75: Entidad Orientación	80
Tabla 4.76: Entidad Sensor de luz.....	80
Tabla 4.77: Entidad Gravedad	80
Tabla 5.1: Columnas de la tabla Usuario	83
Tabla 5.2: Columnas de la tabla Config	84
Tabla 5.3: Columnas de la tabla Sensores	84
Tabla 5.4: Columnas de la tabla Mensajespendientes	84
Tabla 5.5: Columnas de la tabla users	86
Tabla 5.6: Columnas de la tabla dispositivos.....	86
Tabla 5.7: Columnas de la tabla datos_giroscopio	87
Tabla 5.8: Columnas de la tabla datos_velocidad_rotacion	87
Tabla 5.9: Columnas de la tabla datos_presion	87
Tabla 5.10: Columnas de la tabla datos_aceleracion_lineal	88
Tabla 5.11: Columnas de la tabla datos_proximidad	88
Tabla 5.12: Columnas de la tabla datos_acelerometro.....	88
Tabla 5.13: Columnas de la tabla datos_campo_magnetico	88
Tabla 5.14: Columnas de la tabla datos_temperatura	89

Tabla 5.15: Columnas de la tabla datos_orientacion	89
Tabla 5.16: Columnas de la tabla datos_sensor_luz	89
Tabla 5.17: Columnas de la tabla datos_gravedad	89
Tabla 7.1: Especificaciones del terminal móvil de prueba	142
Tabla 7.2: Especificaciones del servidor de prueba	142
Tabla 7.3: Tiempo de ejecución de la monitorización de sensores	¡Error! Marcador no definido.
Tabla 7.4: Prueba de periodo para 1 segundo	143
Tabla 7.5: Prueba de periodo para 2 segundos	143
Tabla 7.6: Prueba de periodo para 3 segundos	144
Tabla 7.7: Tiempo de ejecución del servicio Web	144

1 Introducción

Este documento describirá el análisis y diseño del proyecto titulado “Diseño de un sistema de monitorización de dispositivos móviles”, un proyecto en el que se diseñará y desarrollará una sistema que monitorizarán remotamente los sensores de dispositivo móviles compuesto por una aplicación móvil y una aplicación web.

En esta sección se expondrán los contenidos principales del documento. Se comenzará describiendo el contexto del sistema desarrollado. A continuación se describirán los problemas que se pretenden resolver y los objetivos que se deben alcanzar para ello. Seguidamente se describirá la estructura que presenta el documento. Finalmente se listarán las siglas, acrónimos y el glosario de términos que se utilizarán a lo largo del documento.

1.1 Motivación

Los teléfonos móviles en la actualidad son un elemento indispensable más de las personas junto con las llaves de casa y la cartera con los documentos personales. En el año 2011, la población total de la Tierra superaba la asombrosa cantidad de siete mil millones de personas [1]. En ese mismo año, se calculaba que el número de teléfonos móviles en el mundo era superior a los 5.600 millones [2]. Dentro de pocos años es previsible que el número de dispositivos móviles activos en el mundo supere al número de personas. No es raro encontrar personas que posean un teléfono inteligente y una Tablet, o dos teléfonos, u otros dispositivos móviles. Estos dispositivos móviles almacenan gran cantidad de información sobre su propietario de tal manera que tener control sobre el dispositivo es tener control sobre su dueño.



ILUSTRACIÓN 1.1: DISPOSITIVOS MÓVILES

Los dispositivos móviles cuentan con gran variedad de sensores: acelerómetro, sensor de luz, sensor de temperatura, barómetro, sensor de gravedad, sensor de presión; Sensores que recopilan información de su entorno que puede ser utilizada para comprar la situación del terminal y por tanto del propietario.

Millones de teléfonos son perdidos o robados al año. La pérdida de un teléfono puede suponer, a parte de una pérdida económica, una pérdida de información personal. Como se puede controlar un dispositivo que ya no se encuentra en nuestras manos. Remotamente.

Para controlar un sistema distribuidamente es esencial primero conocer su situación actual. Para averiguar el estado de un dispositivo se puede hacer uso de la información otorgada por los sensores que posee. El desarrollo de este trabajo tiene como objetivo conocer el estado de un terminal a través de los sensores que posee.

1.2 Objetivos

El objetivo principal de este proyecto es **el análisis, el diseño y el desarrollo de un sistema para monitorizar remotamente los valores de los sensores de dispositivos móviles.**

Este sistema estará compuesto por dos subsistemas.

El primer subsistema consistirá en una aplicación móvil para el sistema operativo Android que permita monitorizar los sensores del dispositivo en el que se ejecuta y envíe los datos recopilados al segundo subsistema.

El segundo subsistema consiste en un servidor Web que reciba y almacene los datos enviados por las aplicaciones móviles y los muestre a los usuarios a través de una aplicación Web.

En este documento se podrá seguir el desarrollo de este sistema desde su análisis hasta su implementación y posteriores pruebas.

Para alcanzar este objetivo se han definido una serie de subobjetivos.

1.2.1 Subobjetivos

Los subobjetivos son los listados a continuación:

- Conocer los principios del lenguaje de programación Java para Android
- Diseñar y desarrollar una página web.
- Diseñar y desarrollar una aplicación móvil
- Diseñar e implementar la comunicación entre los subsistemas

1.3 Estructura del documento

En el segundo capítulo del documento se tratará el estado de la cuestión, capítulo donde se tratarán los sistemas relacionados con el sistema que se desarrollará más adelante en el documento. En este capítulo en primer lugar se enumerarán y analizarán los sistemas operativos de los diferentes terminales móviles del mercado: Android, iOS, Windows Phone, Blackberry OS, Symbian y WebOS; A continuación se analizarán aplicaciones de monitorización similares a la desarrollada en este proyecto y los servicios webs como el utilizado para el desarrollo del proyecto.

En el tercer capítulo del documento se tratarán las leyes que afectan al sistema que se va a desarrollar.

En el cuarto capítulo del documento se procederá a realizar un análisis del sistema, se analizarán los casos de uso, los requisitos de usuario y los requisitos de software que presenta el sistema, así como un primer análisis del almacenamiento de los datos.

En el quinto capítulo del documento se procederá a diseñar el sistema utilizando como base de análisis del capítulo anterior. En este capítulo en primer lugar se tratarán los lenguajes de programación que se utilizarán para el posterior desarrollo, se diseñarán las estructuras de las bases de datos, se analizarán las comunicaciones y los servicios necesarios y se diseñarán las interfaces de usuario.

En el sexto capítulo del documento se desarrollará el sistema, Se comenzará desarrollar el sistema. Se comenzará desarrollando los componentes del servidor Web, el servicio y la aplicación. Después de desarrollara la aplicación móvil con su cliente y su aplicación.

En el séptimo capítulo se realizarán y analizaran las pruebas de evaluación sobre el sistema desarrollado.

En el octavo capítulo se expondrán las conclusiones obtenidas de la realización de este proyecto.

En el noveno capítulo se incluirán como anexos los manuales de instalación y de usuario.

Finalmente en el décimo capítulo se listarán las citas bibliográficas utilizadas a lo largo del documento.

1.4 Siglas y Acrónimos

En este apartado se listarán en orden alfabético todas las siglas y acrónimos y sus significados utilizados a lo largo del documento.

TABLA 1.1: SIGLAS Y ACRÓNIMOS

Siglas o acrónimos	Significado
API	Application Programming Interface.
CDMA	Code Division Multiple Access.
CSS	Cascading Style Sheets
GPS	Global Positioning System.
HSPA	High-Speed Packet Access
HDP	Health Device Profile.
HTML	HyperText Markup Language.
HTTP	Hypertext Transfer Protocol.
LTE	Long Term Evolution
NDK	Native Development Kit.
NFC	Near Field Communication.
OS	Operating System.
PC	Personal Computer.
PDA	Personal Digital Assistant.
PHP	Hypertext Pre-processor
QR-Code	Quick Response Code
RAM	Random Access Memory.
SDK	Software Development Kit.
SMS	Short Message Service.
SOAP	Simple Object Access Protocol.
SQL	Structured Query Language.
URI	Uniform Resource Identifier.
URL	Uniform Resource Locator.
W3C	World Wide Web Consortium.
WADL	Web Application Description Language
Wi-Fi	Wireless Fidelity.
WSDL	Web Services Description Language
WSRM	Windows System Resource Manager
XALM	eXtensible Application Markup Language.
XML	eXtensible Markup Language.

1.5 Glosario de términos

En esta sección se listarán alfabéticamente los términos y sus significados utilizados a lo largo del documento.

Código QR o QR-Code: es un sistema para almacenar información en una matriz de puntos o un código de barras bidimensional creado por la compañía japonesa Denso Wave, subsidiaria de Toyota, en 1994. Se caracteriza por los tres cuadrados que se encuentran en las esquinas y que permiten detectar la posición del código al lector.

Framework: es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Tablet: es un tipo de computadora portátil, de mayor tamaño que un teléfono inteligente o una PDA, integrado en una pantalla táctil (sencilla o multitáctil) con la que se interactúa primariamente con los dedos o una pluma stylus (pasiva o activa), sin necesidad de teclado físico ni ratón.

Webkit: es una plataforma para aplicaciones que funciona como base para el navegador web Safari, Google Chrome, Epiphany, Maxthon, Midori, Qupzilla entre otros. Está basado originalmente en el motor de renderizado KHTML del navegador web del proyecto KDE, Konqueror.

2 Estado de la cuestión

En este apartado se analizarán los diferentes sistemas relacionados con el software y hardware utilizados para desarrollar las aplicaciones de este proyecto.

2.1 Sistemas operativos móviles

En esta sección se expondrán la historia y las características de los sistemas operativos más destacados para teléfonos móviles inteligentes [3].

Los sistemas operativos que se analizarán serán:

- Android
- iOS
- Windows Phone
- Blackberry OS
- Symbian WebOS

2.1.1 Android

El primer sistema que se analizará será el sistema operativo Android de Google, sistema utilizado para el desarrollo de la aplicación móvil del proyecto.

2.1.1.1 Historia



ILUSTRACIÓN 2.1: ANDROID LOGO

Android es un sistema operativo basado en Linux destinado a dispositivos móviles desarrollado por la empresa Android Inc., empresa fundada en Palo Alto, California por Andy Rubin, Rich Miner, Nick Sears y Chris White. La empresa fue comprada por Google Inc. en agosto de 2005, tras la compra, Android siguió siendo desarrollado por la Open Handset Alliance, unión de fabricantes de software y hardware como T-Mobile, HTC, Qualcomm, liderada por Google. [4]

Android Software Development Kit fue presentado en noviembre del 2007, pero no fue hasta casi un año después cuando se pudo ver en el mercado el primer teléfono con el sistema operativo Android 1.0, el HTC Dream de la empresa taiwanesa HTC. Esta primera versión estaba integrada con los servicios de Google, su navegador web contaba con zum y era capaz de mostrar varias páginas webs como ventanas, Android Market contaba con opciones de subida y bajada. Android se actualizó a su versión 1.1 el 9 de febrero de 2009, aunque sólo para el dispositivo HTC Dream.

El 30 de abril de 2009 una nueva actualización salió, la versión 1.5 nombrada como Cupcake basada en el kernel de Linux 2.6.27. Esta versión aumentaría la velocidad de inicio y de captura de la cámara de los dispositivos, mejoraría la localización GPS, contaría con la opción de subida directa de vídeos a YouTube y de imágenes a Picassa. Como mayor novedad contaría con un teclado virtual.

El 15 de septiembre del mismo año se liberaría Android 1.6 Donut, basado en el kernel de Linux 2.6.29. Esta versión aumentaría la búsqueda tanto escrita como por voz, incorporaría un indicador de uso de la batería, daría soporte CDMA.



ILUSTRACIÓN 2.2: NEXUS ONE CON ANDROID 2.1

Poco más de un mes después, el 26 de octubre, se lanzaría una nueva actualización, la versión 2.0 Eclair para subsanar los fallos de la versión anterior. Esta versión incorporaría sincronización de contactos de múltiples cuentas, daría soporte a Bluetooth 2.1, contaría con una nueva interfaz de usuario en el navegador y este soportaría HTML5 y el calendario sería ampliado con nuevas características. El último año del 2009 se liberó la SDK 2.0.1, y el primer año del 2010 se actualizó al SDK 2.1.

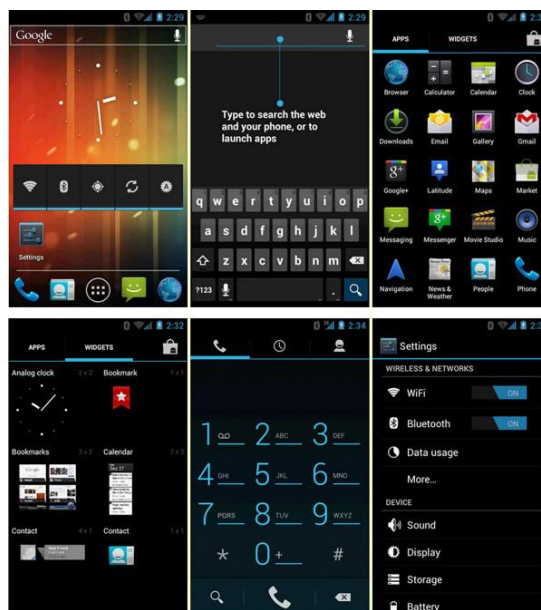


ILUSTRACIÓN 2.3: INTERFAZ DE ICE CREAM SANDWICH

En mayo de 2010, basado en el kernel de Linux 2.6.32 se liberó la versión de Android, 2.2 Froyo. Esta versión incluyó Adobe Flash 10.1, teclados en múltiples idiomas, soporte para punto de acceso.

En diciembre del 2010 se liberó la versión que actualmente es la más distribuida, ilustración 2.4, la versión 2.3 Gingerbread. Versión que sería actualizada a su versión 2.3.3 en febrero del siguiente año. Esta versión incorporaría llamadas a través de internet, NFC, nuevo teclado, nuevas formas de copiado y pegado de texto.

Con el aumento del número de tablets en el mercado, Google presentó la primera versión de su sistema operativo para móviles optimizado para tablets, la versión 3.0 Honeycomb.

En Octubre del año 2011, Google liberaría la versión 4.0 Ice Cream Sandwich, versión única tanto para móviles como para tablets. Esta versión contaría con una nueva tipografía llamada Roboto, una nueva forma de desbloqueo mediante lectura de gestos de la cara, soporte Bluetooth HDP y Wi-Fi Direct.

En la inauguración de Google I/O 2012 celebrada en el mes de junio, se presentó la versión 4.1 Jelly Bean. El primer dispositivo que contaría con esta nueva versión sería la primera Tablet de Google, la Nexus 7, ilustración 2.5, fabricada por Asus. Los dispositivos Galaxy Nexus, ilustración 2.5, y Nexus S se convertirían en los primeros móviles que se actualizarían a esta versión. Jelly Bean mejoraría la fluidez y rapidez de Ice Cream Sandwich, mejoraría entre otras aplicaciones [5] el rendimiento y la compatibilidad del HTML5 con el navegador.

En el gráfico 2.4 y en la tabla 2.4 [6] se pueden observar la actual distribución de versiones del sistema operativo proporcionada por Google mes a mes.

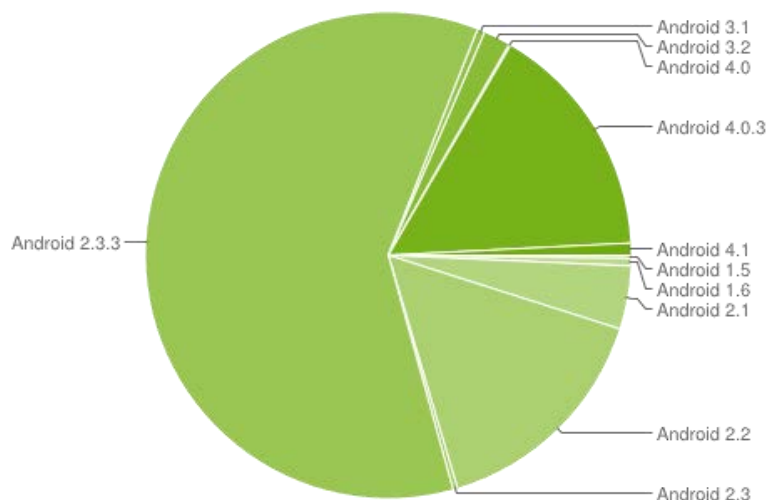


ILUSTRACIÓN 2.4: DISTRIBUCIÓN ACTUAL DE VERSIONES

Como se puede ver el mayor problema del sistema operativo es la fragmentación, esto se deba a que la compañía deja en manos de los fabricantes la actualización de sus dispositivos, es por esto que muchos dispositivos pueden tardar o no llegar a actualizarse a las nuevas versiones del sistema.

TABLA 2.1: DISTRIBUCIÓN ACTUAL DE VERSIONES ANDROID

Versión	Nombre	API	Distribución
Android 1.5	Cupcake	3	0.2%
Android 1.6	Donut	4	0.5%
Android 2.1	Eclair	7	4.2%
Android 2.2	Froyo	8	15.5%
Android 2.3 – 2.3.2	Gingerbread	9	0.3%
Android 2.3.3– 2.3.7		10	60.3%
Android 3.1	Honeycomb	12	0.5%
Android 3.2		13	1.8%
Android 4.0 – 4.0.2	Ice Cream Sandwich	14	0.1%
Android 4.0.3– 4.0.4		15	15.8%
Android 4.1	Jelly Bean	16	0.8%

DATOS RECOGIDOS DURANTE UN PERIODO DE 14 DÍAS QUE FINALIZÓ EL 1 DE AGOSTO DE 2012

2.1.1.2 Características

Android es un sistema operativo de código abierto, el código fuente está disponible bajo licencias de software libre.

El desarrollo de aplicaciones para el sistema operativo Android se realiza principalmente mediante su SDK y el lenguaje de programación Java. Otra opción es utilizar el lenguaje C utilizando el NDK de Google. Se puede utilizar el entorno de desarrollo Eclipse junto con las librerías proporcionadas por Google para el desarrollo de aplicaciones con este software.

2.1.1.2.1 Características principales

A continuación se citarán las principales características de un dispositivo Android. [7]

Framework de aplicaciones: permite reutilizar y remplazar componentes.

Máquina virtual Dalvik: optimizada para dispositivos móviles.

Navegador integrado: basado en el motor de código abierto Webkit.

Gráficos optimizados: alimentado por una biblioteca de gráficos 2D, los gráficos 3D están basados en la especificación OpenGL ES 1.0.

SQLite: para almacenamiento de datos estructurados.

Soporte de medios: para formatos comunes de audio, vídeo e imágenes (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).

Telefonía GSM: depende del hardware del dispositivo.

Bluetooth, EDGE, 3G y Wi-Fi: depende del hardware del dispositivo.

Cámara, GPS, brújula, acelerómetro: depende del hardware del dispositivo.

Ambiente rico de desarrollo: Incluye emulador del dispositivo, herramientas de depuración, perfiles de memoria y rendimiento, y un complemento para Eclipse IDE.



ILUSTRACIÓN 2.5: GALAXY NEXUS Y NEXUS 7 CON ANDROID 4.1

2.1.1.2.2 Arquitectura

En la ilustración 2.6 [7] se pueden observar los diferentes componentes del sistema operativo Android divididos por capas.

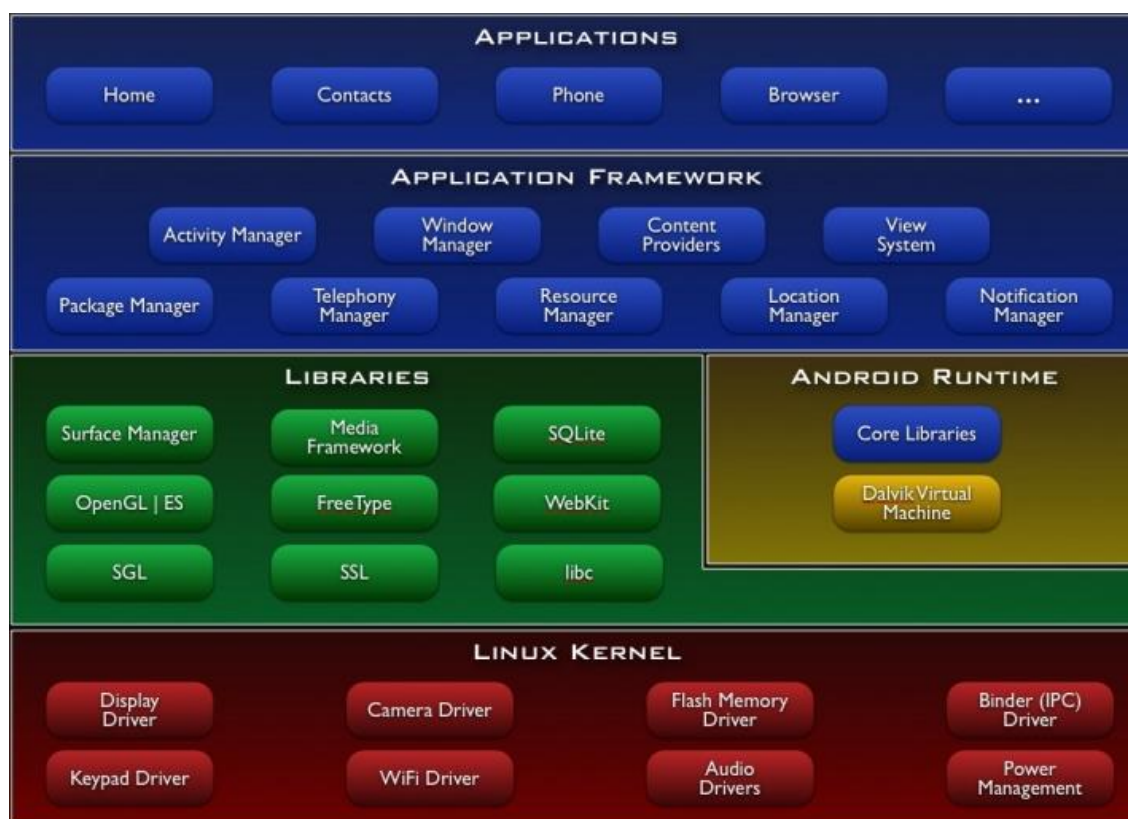


ILUSTRACIÓN 2.6: ARQUITECTURA ANDROID

Aplicaciones.

Todo teléfono Android contará con un conjunto de aplicaciones básicas que incluyen entre otras un cliente de correo electrónico, programa de SMS, calendario, mapas, buscador, contactos. Todas las aplicaciones han sido programadas utilizando el lenguaje Java.

Marco de aplicaciones.

Android ofrece a los desarrolladores la capacidad de crear aplicaciones muy ricas e innovadoras. Los desarrolladores tienen un acceso total al mismo API del framework utilizado por las aplicaciones básicas. La arquitectura está diseñada para simplificar la reutilización de componentes.

Bibliotecas.

Android incluye un conjunto de bibliotecas C/C++ utilizados por diversos componentes del sistema operativo. Estas capacidades se exponen a los desarrolladores a través del marco de aplicaciones. Entre estas bibliotecas se encuentran: System C library, 3D library, SQLite.

Android runtime.

Android incluye un conjunto de bibliotecas que proveen de la mayor parte de las funcionalidades disponibles en las bibliotecas básicas del lenguaje Java.

Kernel de Linux.

Android está basado en el sistema operativo Linux en su versión 2.6, algunos de los servicios básicos como la seguridad, la gestión de memoria, la gestión de procesos, la pila de red; basan su funcionamiento en esto. El núcleo actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

2.1.1.3 Componentes de una aplicación

Dado que la aplicación que se desarrollará en este proyecto se desarrollará para el sistema operativo Android en este punto se describirán los principales componentes que forman una aplicación Android, los Activity y los Service, para ayudar a comprender mejor el diseño y el desarrollo de la aplicación que se comentarán en puntos más adelante.

2.1.1.3.1 Activity

Es el componente más utilizado en las aplicaciones Android. Un componente Activity refleja una determinada actividad llevada a cabo por una aplicación. La mayoría de las ocasiones un Activity está asociado a una interfaz de usuario, dando funcionalidades a los componentes de la interfaz.

En la ilustración 2.7 se puede apreciar el ciclo de vida de un Activity [8].

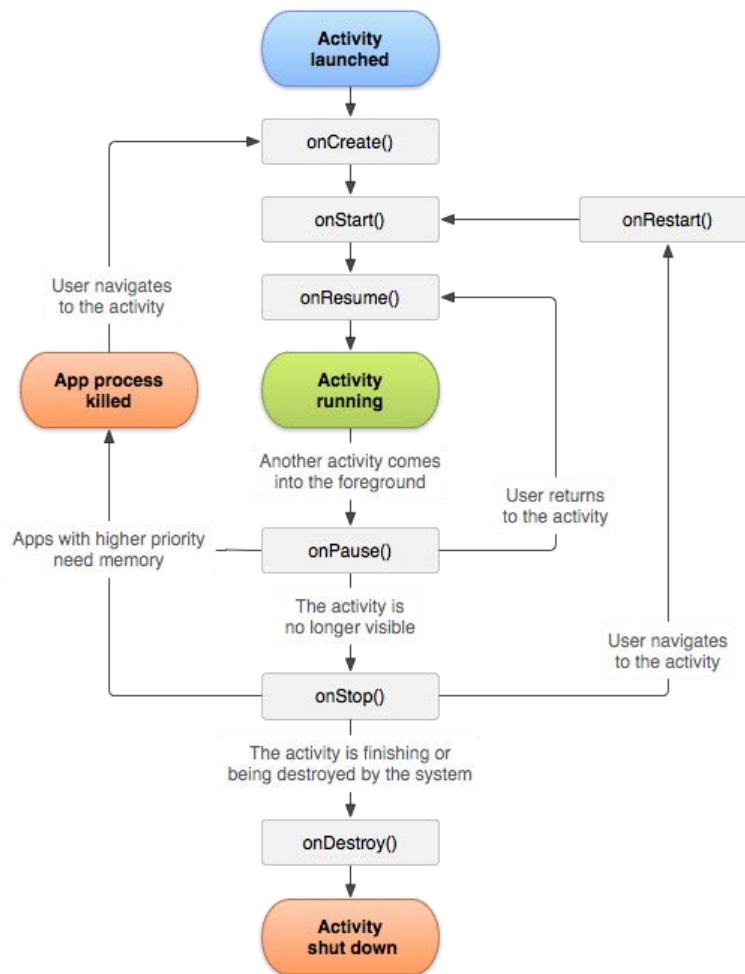


ILUSTRACIÓN 2.7: CICLO DE VIDA DE UN ACTIVITY

Como se puede observar al iniciarse un Activity lo primero que se ejecutará será el código situado en la función `onCreate()`, tras esta se ejecutará el código de la función `onStart()`, desde esta función el Activity ya será visible por el usuario. En caso de que un segundo Activity pase de segundo a primer plano hará que el Activity actual ejecute la función `onPause()`, al volver al primer plano ejecutará `onResume()`. Cuando el Activity deje de ser visible para el usuario se ejecutará el código situado en la función `onStop()`. En caso de que el Activity termine o sea destruido se ejecutará en último lugar `onDestroy()`.

2.1.1.3.2 Service

En un componente Service se ejecuta principalmente una tarea en la cual no participa el usuario, no cuenta con interfaz y se ejecuta en segundo plano mientras otras aplicaciones se encuentran en primer plano incluso si el usuario no está interactuando directamente con la aplicación.

2.1.1.3.3 Broadcast Intent Receiver

Un componente Broadcast Intent Receiver tiene como finalidad lanzar alguna ejecución dentro de una aplicación cuando se produzca un determinado evento. En la mayoría de las ocasiones se utiliza para mostrar notificaciones de los eventos que ocurren en el dispositivo. Entre los principales eventos que puede capturar se encuentran:

- `android.provider.Telephony.SMS_RECEIVED`: Evento de mensaje recibido.
- `android.intent.action.PHONE_STATE`: Evento de llamadas recibidas.
- `android.intent.action.AIRPLANE_MODE`: Evento modo vuelo.
- `android.intent.action.BATTERY_LOW`: Evento batería baja.
- `android.intent.action.BOOT_COMPLETED`: Evento de inicio del sistema operativo.
- `android.intent.action.SCREEN_OFF`: Evento bloqueo de pantalla.
- `android.intent.action.SCREEN_ON`: Evento desbloqueo de pantalla.
- `android.bluetooth.intent.action.DISCOVERY_STARTED`: Evento comienzo de escáner Bluetooth.
- `android.bluetooth.intent.action.ENABLED`: Evento Bluetooth habilitado.

2.1.2 iOS



ILUSTRACIÓN 2.8: iOS LOGO

iPhone Operative System, más tarde conocido como iOS, es el sistema operativo de los dispositivos móviles de la compañía Apple derivado de Mac OS X, el sistema operativo de los Macbooks.

2.1.2.1 Historia

La compañía estadounidense Apple presentó el 9 de junio del año 2007 su primer teléfono inteligente, el iPhone, este teléfono contaría también con el primer sistema operativo para dispositivos móviles de la compañía llamado iPhone OS. Poco después, en septiembre del mismo año se sumaría un nuevo dispositivo a este sistema operativo, el iPod Touch.

La segunda versión de este sistema operativo se introdujo con la presentación del segundo terminal de Apple, el iPhone 3G. La tercera versión tardaría un año más incluyéndose en el siguiente teléfono de Apple, el iPhone 3GS. No sería hasta la cuarta versión, que se incluiría con el iPhone 4 en 2010, cuando el sistema operativo sería bautizado finalmente como iOS. En este año la compañía presentaría el iPad, un nuevo dispositivo tablet que se sumaría a la familia de dispositivos que usarían el sistema iOS.

La última versión presentada es la versión iOS 5 que se dio a conocer con el dispositivo iPhone 4S en el año 2011. La próxima versión del sistema operativo, iOS 6 estará disponible el próximo otoño [9].

2.1.2.2 Características

El desarrollo de aplicaciones para el sistema operativo iOS se realiza con el lenguaje de programación Objective-C, un lenguaje de programación orientado a objetos creado como un superconjunto de C. Las aplicaciones han de ser escritas y compiladas para la arquitectura ARM.

En marzo de 2008 se liberó el SDK para los desarrolladores que, junto al software Xcode, les permitiría desarrollar aplicaciones para iPhone, iPod touch e iPad.

2.1.2.2.1 Características principales

El sistema operativo de Apple al ser un sistema diseñado para un único dispositivo, cuenta con un software muy optimizado para el hardware en el que se ejecuta.



ILUSTRACIÓN 2.9: IPHONE 4S CON iOS 5

Home Screen: La interfaz de usuario está compuesta por una pantalla de inicio deslizable lateralmente, homescreen, que muestra los iconos de las aplicaciones del dispositivo.

Multitasking: Hasta la cuarta versión del sistema, la multitarea estaba reservada a aplicaciones del propio sistema. Desde iOS4 las aplicaciones de terceros soportaban las APIs para la multitasking.



ILUSTRACIÓN 2.10: INTERFAZ DE iOS

Todos los dispositivos iOS cuentan con cámara, bluetooth, 3G, Wi-Fi, GPS, dan soporte de medios: para formatos comunes de audio, vídeo e imágenes (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF) y utilizan SQLite para el almacenamiento de datos estructurados.

Flash: iOS no soporta la tecnología Flash por ser considerado inseguro. Esto afecta principalmente a la navegación web donde es común ver webs que utilizan Flash para mostrar su contenido.

2.1.2.2.2 Arquitectura

En la ilustración 2.11 [10] se pueden observar las capas de la arquitectura del sistema operativo iOS.

En un primer nivel se sitúa la capa de aplicaciones, también conocida como Cocoa Touch, un API del sistema operativo que proporciona una capa de abstracción para crear aplicaciones. En la capa Core Service se encuentran las API de seguridad del sistema basadas en los servicios de la capa Core OS, la capa del núcleo del sistema operativo.



ILUSTRACIÓN 2.11: ARQUITECTURA IOS.

2.1.3 Windows Phone



Windows® Phone

ILUSTRACIÓN 2.12: WINDOWS PHONE LOGO

Microsoft, líder de los sistemas operativos en PC y ordenadores portátiles, se introdujo en los dispositivos móviles con un sistema operativo que tardó en comenzar a hacerse un hueco en

el mercado de los sistemas operativos para móviles.

2.1.3.1 Historia

Windows Phone es un sistema operativo desarrollado por la multinacional Microsoft. Es el sucesor del sistema operativo Windows Mobile, un sistema operativo diseñado para el uso en smartphones que surgió como una ramificación de Windows CE, originalmente fue nombrado como Pocket PC.

Pocket PC fue presentado en el año 2000. En el año 2001 se integro con MSN Messenger y Media Player 8 Enhanced UI. Dos años después, en el año 2003, Pocket PC se convirtió en Windows Mobile con integración de Bluetooth y Pocket Internet Explorer Windows Media Player 9.0. En el año 2004 dio soporte a la conmutación entre pantalla horizontal, landscape, y pantalla vertical, portrait para Pocket PCs y



ILUSTRACIÓN 2.13: NOKIA LUMIA 900 CON WINDOWS PHONE 7

WPA. Windows Media Player 10 Mobile, interfaz GPS e introducción a office mobile fueron introducidos en el 2005. En el año 2009 en la versión Windows Mobile 6.5 el sistema integro Internet Explorer Mobile 6 y soporte Multitouch.

Windows Phone 7 fue presentado el 15 de febrero de 2010 en el Mobile World Congress en Barcelona. La versión final, Windows Phone 7 se lanzó el 1 de septiembre del mismo año.

En otoño de 2012 estará disponible la nueva versión del sistema operativo, Windows Phone 8, entre sus novedades mejorará el soporte para procesadores multinúcleo que podrá gestionar teóricamente hasta 64 núcleos, soporte para mejor resolución de pantalla que permitir-a gestionar resoluciones de hasta WXGA, soporte para tarjetas MicroSD, soporte nativo para NFC Secure.

2.1.3.2 Características

Windows Phone está basado en el núcleo del sistema operativo Windows Embedded CE 6.0.

El desarrollo de aplicaciones se puede realizar utilizando Microsoft Silverlight o Microsoft XNA Framework. La primera permite el desarrollo de aplicaciones basadas en XAML, incluye Base Class Library, un conjunto de clases que dan soporte a la lectura y escritura de ficheros, manipulación de archivos XML y manejo de gráficos. La segunda es una implementación nativa de .NET Compact Framework que incluye gran variedad de bibliotecas de clases, específicos para el desarrollo de juegos, tratamiento de dispositivos de entrada, sonidos, vídeos.

2.1.3.2.1 Características principales

Las características y los servicios de Windows Phone pueden diferir en función de varios factores: idiomas, configuración regional, operadora, fabricante del Hardware, modelo.

Windows Phone está diseñado para dispositivos con pantallas táctiles, la pantalla de inicio se compone de mosaicos dinámicos personalizables que muestran el icono de la aplicación e información.

El sistema operativo utiliza para todas las búsquedas el motor de búsqueda Bing de Microsoft y como navegador web Internet Explorer 9.

Soporte de redes Wi-Fi, Bluetooth, 3G; Y servicio de Xbox LIVE.

2.1.3.2.2 Arquitectura

En la ilustración 2.14 [11] se puede observar la arquitectura de Windows Phone, con framework para el desarrollo de aplicaciones para el sistema y controladores para los diferentes tipos de hardware que tenga el dispositivo.

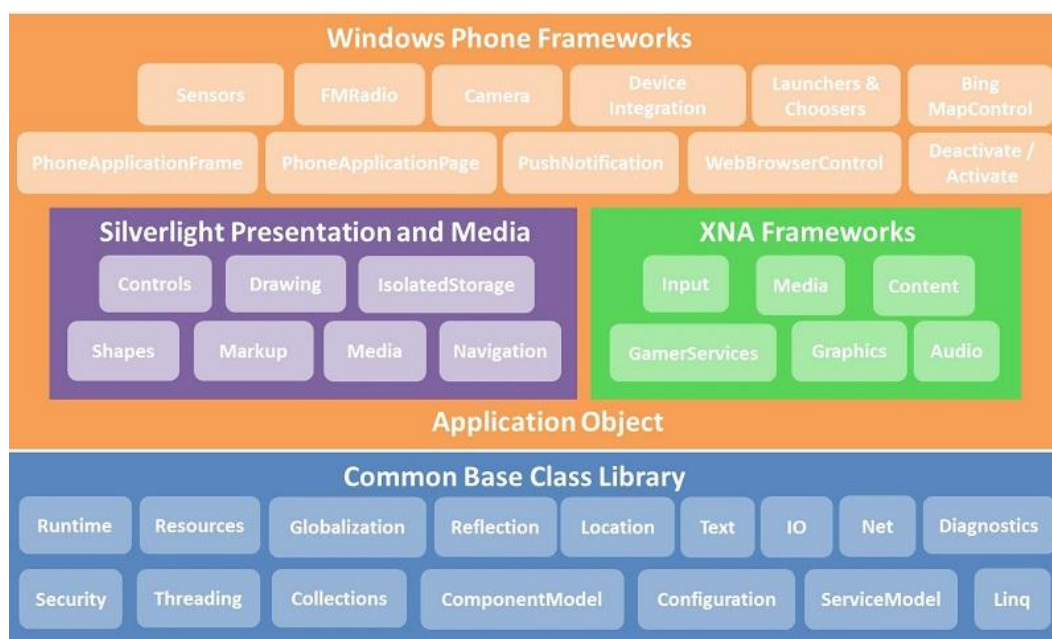


ILUSTRACIÓN 2.14: ARQUITECTURA WINDOWS PHONE

2.1.4 Blackberry OS

ILUSTRACIÓN 2.15:
BLACKBERRY LOGO

Blackberry OS es un sistema operativo para dispositivos móviles desarrollado por Research in Motion, empresa canadiense más conocida como RIM, que sirve como sistema operativo de los dispositivos de la misma empresa también denominados Blackberry.

2.1.4.1 Historia

El primer dispositivo con el que se presentó RIM data de 1999, fue el busca 850 que ofrecía un calendario, organizador, correo. A partir de la serie 5000 y 6000 La compañía comenzó cosechar éxito mundial. La serie 7200 fue el primer terminal de la compañía con pantalla a color, la 7250 fue el primer modelo con Bluetooth, el modelo 7270 fue el primero con Wi-Fi y el 7520 integraba GPS. La serie 8700 que dieron a conocer en noviembre de 2005 sirvió a RIM a alcanzar los 5 millones de usuarios en marzo de 2006.

En 2010 RIM presentó Blackberry 6, en este sistema la compañía se centró en la parte multimedia enfocada al usuario, integrando redes sociales y la mensajería instantánea.

Blackberry lanzó su primera Tablet el 19 de abril de 2011, la Blackberry PlayBoook, el sistema operativo se modificó para adaptarlo a estos terminales, se denominó Blackberry Tablet OS basado en QNX Neutrino, un sistema operativo de tiempo real basado en Unix. Según RIM, QNX podría ser incluido en la próxima generación de Blackberry 10 OS.

2.1.4.2 Características

La versión actual del sistema operativo es la versión 7.1. El sistema operativo se diseñó originalmente para un uso profesional, se buscó la seguridad y la privacidad, por ello el sistema ofrece acceso a las cuentas de correo electrónico, agenda, libreta de direcciones, eventos, sincronización y otras funcionalidades mediante programas externos.

2.1.4.2.1 Características principales

La principal característica e identificación de los terminales BlackBerry es el teclado completo QWERTY en la mayoría de sus terminales. Las actuales versiones del sistema operativo dan soporte a la multitarea y a diferentes sistemas de entrada a parte del teclado como pantallas táctiles, touchpad, trackball.

La seguridad que ofrece está condicionada a suscripciones a servicios asociados de mensajería y gestión de correo [12]. BlackBerry Enterprise Server, BES, es el servicio asociado para empresas y profesionales y usuarios que buscan un gran nivel de seguridad. Entre sus principales funciones se encuentra el control total del buzón de entrada e identificación de los correos, gestión de los filtros seleccionados por los propios usuarios y comprime y optimiza los mensajes, encriptándolos y enviándolos a los dispositivos asociados a la cuenta. BlackBerry Internet Service, BIS, es el sistema indicado para usuarios particulares, permite el uso de la tecnología PUSH y se pueden sincronizar hasta diez cuentas de correo.

BlackBerry OS también cuenta con un sistema de entretenimiento que incluye BlackBerry Music, un reproductor de música, y BlackBerry Messenger, un sistema de mensajería instantánea entre dispositivos BlackBerry.



ILUSTRACIÓN 2.16: BLACKBERRY BOLD 9790 CON BLACKBERRY OS 7

2.1.5 Symbian



ILUSTRACIÓN 2.17: SYMBIAN LOGO

Symbian OS es un sistema operativo creado por una alianza de varias empresas de telefonía móvil para dispositivos móviles con la idea original de competir con los sistemas operativos Palm y Windows Mobile.

2.1.5.1 Historia

Symbian fue creado por la alianza entre Nokia, Sony Ericsson, Samsung, Siemens, Fujitsu, Motorola, Lenovo, LG, Mitsubishi Electric, Panasonic y otras. Proviene de EPOC, un sistema operativo que data de finales de los años ochenta y principios de los noventa como EPOC16, en 1997 apareció EPOC32 que más tarde sería conocido como Symbian.

En el año 2000 salió el primer teléfono Symbian, el Nokia 9210. En 2004 Psion vendió su participación en Symbian LTD, ese mismo año apareció el primer gusano del sistema, utilizaba el Bluetooth para dispersarse entre los teléfonos. Symbian OS 8.0 daba la opción de elegir entre dos núcleos, los kernels EKA1 y EKA2. Symbian 9.0 se utilizó exclusivamente con propósitos internos del sistema operativo, se dejó de fabricar en 2004 y marcó el final del kernel EKA1. La actualización a la versión 9.1 incluyó varias características de seguridad, la versión 9.3 incluyó mejoras en la gestión de memoria y soporte nativo para Wi-Fi 802.11, HSDPA. En 2006 se anunció la versión 9.4 diseñado para redes 2.5G y 3G.

2.1.5.2 Características

Symbian OS posee un núcleo de tiempo real, es un sistema operativo basado en un microkernel y con capacidad multireading [13]. Es un sistema estructurado con multitarea y protección de memoria.

2.1.5.2.1 Características principales

Al ser diseñado para dispositivos portátiles, dispositivos con capacidades limitadas, memoria y otros recursos, está muy optimizado para sacar el máximo aprovechamiento al dispositivo.

Cada aplicación se ejecuta en sus propios procesos y tienen acceso únicamente a su propio espacio de memoria, debido a esto las aplicaciones para Symbian están orientadas a un hilo.

El sistema operativo cuenta con una base de datos SQL, seguridad y soporte para varias plataformas de desarrollo como C, C++, Java ME. Toda la programación está basada en eventos.

Las últimas versiones soportan paginación bajo demanda que mejora el aprovechamiento de la memoria RAM ya que sólo se carga en memoria la página a ejecutar.

2.1.5.2.2 Arquitectura

En la ilustración 2.19 se puede observar la arquitectura del sistema operativo Symbian [14]. Los servicios de seguridad se desarrollan en múltiples niveles. El servicio de mensajería, el buscador y las aplicaciones multimedia son ejemplos de subbloques dentro de los servicios de aplicación.



ILUSTRACIÓN 2.18: NOKIA N8
CON SYMBIAN^3

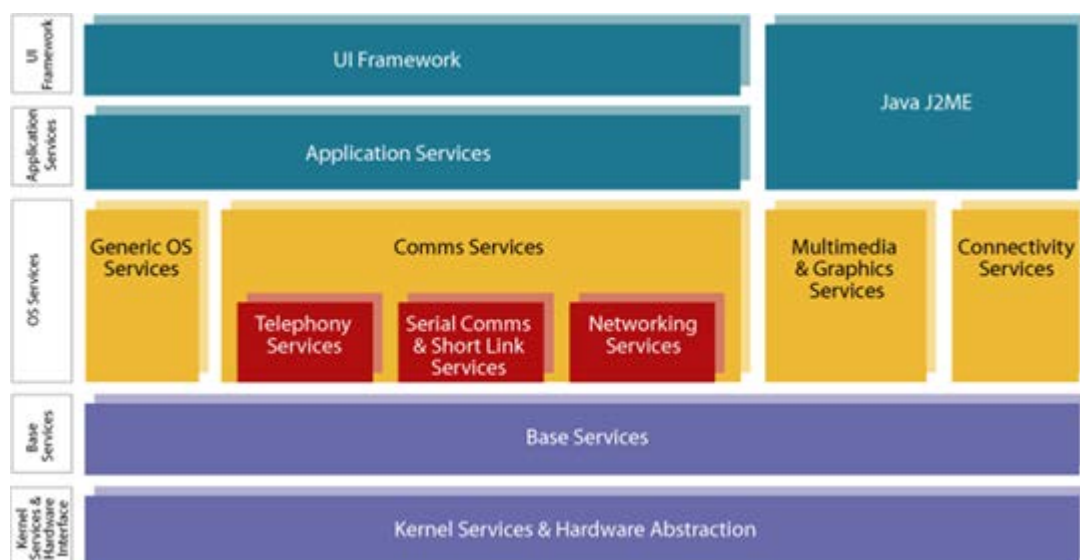


ILUSTRACIÓN 2.19: ARQUITECTURA SYMBIAN

2.1.6 WebOS



WebOS es un sistema operativo basado en Linux que se creó originalmente para servir como sistema operativo para PDAs por Palm Inc. como Palm OS, actualmente es propiedad de Hewlett-Packard Company, más conocida como HP.

ILUSTRACIÓN 2.20: HP WEBOS LOGO

2.1.6.1 Historia

El desarrollo del sistema Palm OS se remonta a 1996, fue registrado en diciembre de 1997 por Palm Inc. Las primeras versiones del sistema estuvieron basadas en un sistema operativo multitarea creado por Motorola.

La versión 1.0 del sistema se vendía con los primeros Pilot 1000 y 5000. La versión 2.0 se introdujo con Palm Pilot Personal y Profesional. La versión 3.0, 3.1, 3.3 y 3.5 incluían soporte de color, puertos de expansión, nuevos procesadores. Con la serie m500 salió la versión 4.0, se añadió una interfaz estándar para acceso a sistemas de archivos externos y mejoraba las bibliotecas de telefonía y seguridad. La versión 5 del sistema fue la primera que soportó los dispositivos ARM como iniciativa de apoyar a estos procesadores.

En enero del año 2009 Palm presentó WebOS como sucesor de Palm OS, en junio del mismo año Palm Pre se convertiría en el primer dispositivo en utilizar este sistema operativo.



ILUSTRACIÓN 2.21: HP TOUCHPAD CON WEBOS 3

En abril del año 2010 HP adquiriría Palm por 1.200 millones de dólares. HP anunciaría su intención de proseguir con el desarrollo de WebOS ampliándolo a nuevos dispositivos como las tablets.

En febrero de 2011 HP anunciaría su intención de utilizar WebOS como plataforma universal para todos sus dispositivos. El 18 de agosto de HP comunicó que no continuarían fabricando dispositivos con WebOS. El 9 de diciembre HP anunció que continuaría dando soporte al sistema operativo aunque desde entonces se convertiría en software libre.

La última versión del sistema es la versión 3.0.5 que data de enero de 2012.

2.1.6.2 Características

WebOS es un sistema creado para dispositivos con pantalla táctil, incluye un conjunto de aplicaciones para el manejo de información personal y utiliza lenguajes web como HTML5, CSS y JavaScript.

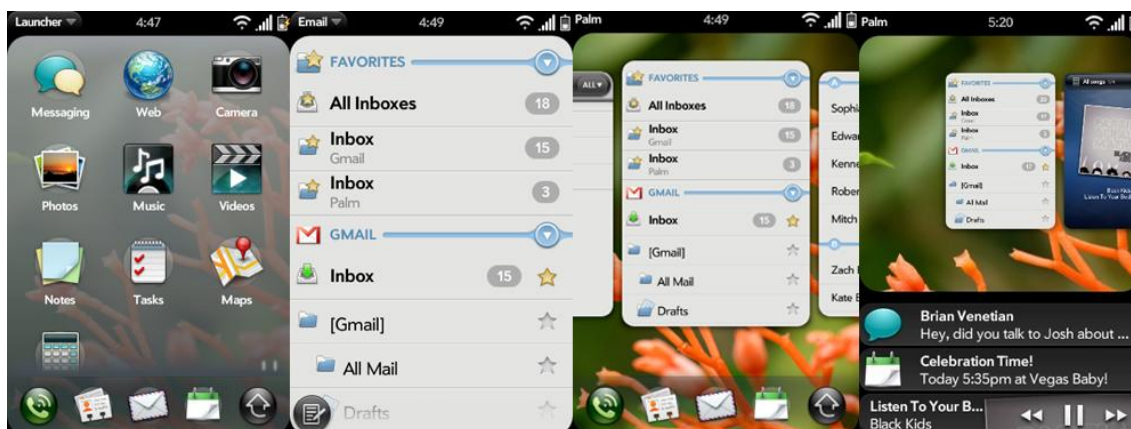


ILUSTRACIÓN 2.22: INTERFAZ PALM WEBOS

2.1.6.2.1 Características principales

Palm OS es un sistema operativo utilizado en variedad de dispositivos: smartphones, relojes, PDAs, GPS; desde la versión 5.0 este sistema operativo se ejecuta en procesadores basados en la arquitectura ARM.

El sistema contaba con Synergy, una característica que permitía agrupar información de varias cuentas personales del usuario en una sola fuente en el dispositivo.

La actualización a WebOS 2.0 contó con Stacks, que permitía que los usuarios interactuaran con las aplicaciones como si fueran cartas, Just Type, el buscador universal de WebOS tanto para web como para los archivos internos, Exhibition, un protector de pantalla, soporte para funciones HTML5.

La versión 3.0.5 añadía mejoras en las aplicaciones básicas como la del correo electrónico y el calendario, mejoraba la experiencia de video llamadas y corregía errores de versiones anteriores.

2.1.6.2.2 Arquitectura

Las ilustraciones 2.21 y 2.22 reflejan la arquitectura del sistema operativo WebOS [15].

El sistema operativo está basado en el kernel de Linux 2.6 combinado con componentes de código abierto de Palm, es compatible con el sistema de ficheros ext3 para las particiones de ficheros internas y fat32 para las particiones de ficheros externas. El usuario no tendrá ningún tipo de interacción directa con el sistema operativo básico, la interacción se realizará a través de Mojo, varias aplicaciones y la UI System Manager, responsable de la interfaz de usuario.

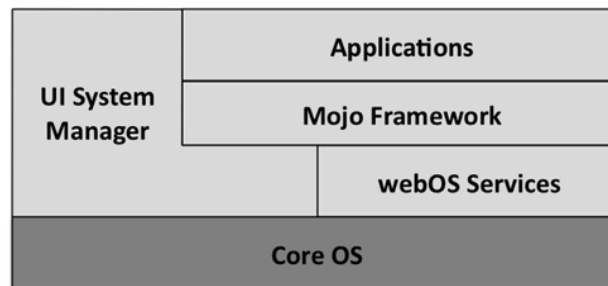


ILUSTRACIÓN 2.23: ARQUITECTURA WEBOS SIMPLIFICADA

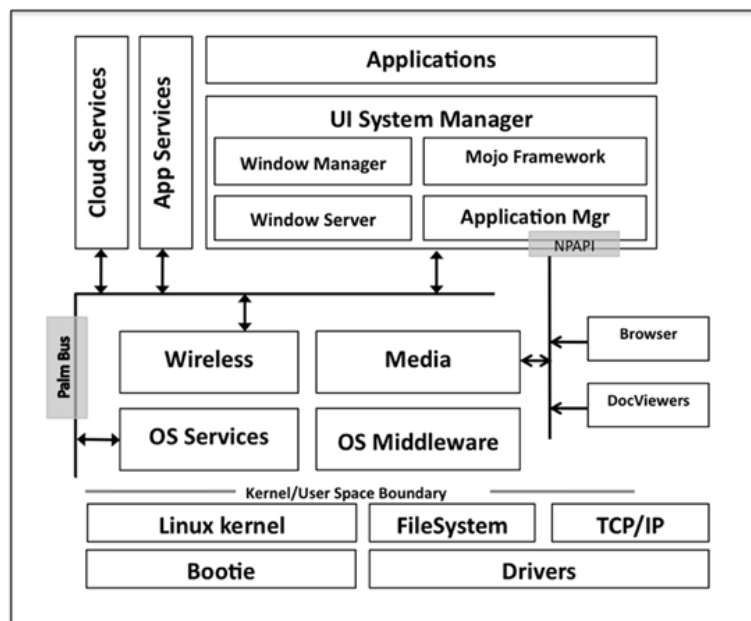


ILUSTRACIÓN 2.24: ARQUITECTURA WEBOS

2.2 Servicios web

Los Servicios Web son sistemas software diseñados para soportar una interacción interoperable máquina a máquina sobre una red.

En este apartado se analizarán y compararán REST y SOAP y se analizarán algunos frameworks basados en REST.

2.2.1 REST

REST, Representation State Transfer o transferencia de estado representacional, es un estilo de arquitectura de software para sistemas distribuidos tales como la Web. El término fue introducido en la tesis doctoral de Roy Fielding en el año 2000.

REST se refiere a una colección de principios para el diseño de arquitecturas en red. El término es utilizado frecuentemente en el sentido de describir a cualquier interfaz que transmite datos específicos de un dominio HTTP sin una capa adicional como hace SOAP. REST no es un estándar, es un estilo de arquitectura basado en estándares como HTTP, URL, XML, HTML, MIME.

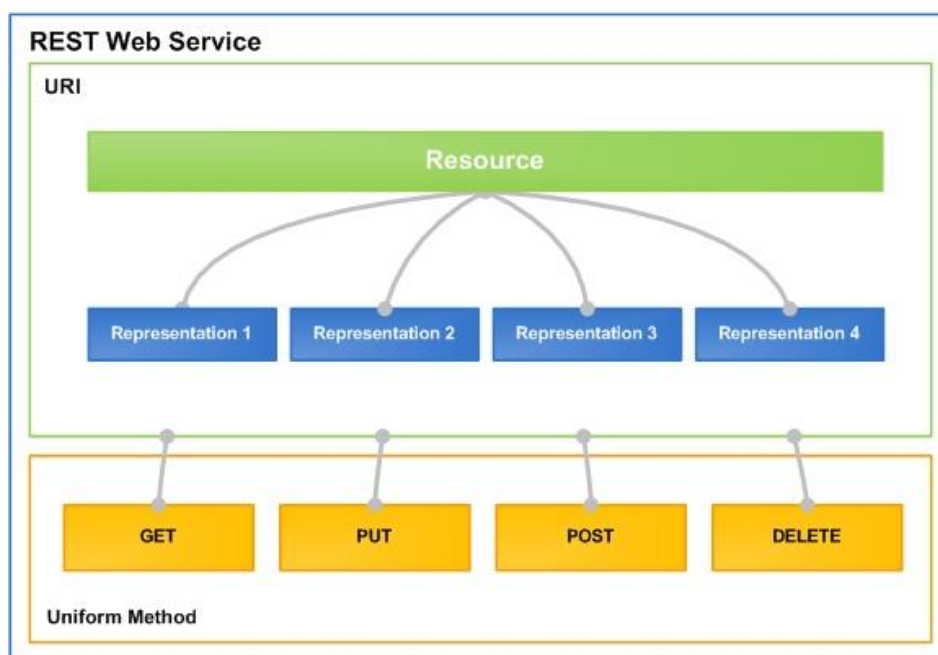


ILUSTRACIÓN 2.25: SERVICIO WEB REST

Los principios de REST son:

Escalabilidad de la interacción con los componentes. La Web ha crecido exponencialmente sin degradar su rendimiento.

Generalidad de interfaces. Gracias al protocolo HTTP, cualquier cliente puede interactuar con cualquier servidor HTTP sin ninguna configuración especial.

Puesta en funcionamiento independiente. Los servidores y los clientes pueden estar ejecutándose durante años, por lo que los servidores antiguos deben de ser capaces que comunicarse con los más actuales clientes y viceversa.

Compatibilidad con componentes intermedios. Como los proxies, las cachés, que se utilizan para mejorar el rendimiento, los firewalls, que permiten mejorar las políticas de seguridad.

REST cumple estos principios con ciertas restricciones:

Identificación y manipulación de recursos. Esto se consigue mediante el uso de URIs. Los recursos son los objetos lógicos a los que se le envían los mensajes. Los recursos no pueden ser directamente accedidos o modificados, se trabaja con una representación de estos.

Mensajes autodescriptivos. REST dicta que los mensajes HTTP deberían ser lo más descriptivamente posibles. Gracias a esto los intermediarios interpretan los mensajes y ejecutan servicios en nombre del usuario.

Hipermedia como un mecanismo del estado de la aplicación. El estado actual de una aplicación Web debería ser capturado en uno o más documentos de hipertexto, residiendo tanto en el cliente como en el servidor.

2.2.1.1 Axis



ILUSTRACIÓN 2.26: AXIS LOGO

Apache Axis es una implementación de SOAP, Simple Object Access Protocol, que proporciona un entorno de ejecución para Servicios Web implementados en Java. Axis se desarrolla bajo supervisión de la Apache Software Foundation. La implementación de SOAP es tanto Java como C, utiliza varias APIs para generar y utilizar los servicios web. Axis implementa JAX-RPC y provee de un contenedor versátil.

Con Apache Axis se pueden crear aplicaciones de computación distribuida que utilicen servicio web localizado en sistemas heterogéneos, exponer servicios en aplicaciones web existentes o crear un servidor dedicado que ofrezca servicios.

2.2.1.2 Axis2



ILUSTRACIÓN 2.27: AXIS2 LOGO

Apache Axis2 es un rediseño y una reimplementación total del contenedor de Web Services Axis [16]. Esta segunda versión de Axis evolucionó independientemente de la primera versión debido a que implementa especificaciones diferentes.

Axis2 implementa la especificación JAX-WS del Java Community Process, WS-Messging y WS-Security, dispone de una estructura modular que permite

ampliar la funcionalidad básica del sistema en el futuro con módulos adicionales al estilo plugins.

Soporta protocolos de Web Services más actuales que SOAP como REST.

Todas las funcionalidades de Axis2 se encuentran agrupadas en módulos, se distinguen los módulos básicos y los módulos opcionales. Cada módulo tiene asociados una serie de flujos de entrada y de salida sobre los que se puede configurar handlers o manejadores, estos handlers la mayoría de las ocasiones serán clases que el contenedor notificará cuando se produzca un evento.

Uno de los elementos principales es AXIOM, Axis, Object Model, es un modelo de objetos que se emplea para serializar y deserializar mensajes SOAP usando las APIs de la familia JAX, Java Api for XML. AXIOM aporta su capa de abstracción propia por encima de las APIs de JAX.

2.2.1.3 Restlet



ILUSTRACIÓN 2.28: RESTLET LOGO

Restlet es un completo y ligero framework creado por Jerome Louvel en 2007 que sirve para desarrollar clientes y servicios REST en la plataforma Java [17].

Restlet da soporte a los conceptos de estilo de arquitectura Transferencia de Estado Representacional, REST, recursos, representación, conectores y componentes.

Restlet soporta los principales estándares Web como HTTP, SMTP, XML, JSON, OData, RDF, RSS, WADL y Atom. Cuenta con diversas extensiones para integrarse con Servlet, Spring, Jetty, Grizzly, Simple, JAXB, JIBX, Velocity, FreeMarker y más.

Restlet está disponible en cinco ediciones. Cada edición se dirige a un entorno de desarrollo especial:

- Google Web Toolkit (GWT) para aplicaciones AJAX desplegados en los navegadores de escritorio sin necesidad de plug-in extras.

- Google App Engine (GAE/J) para el despliegue en infraestructuras de computación en la nube de Google.
- Android para el despliegue en teléfonos inteligentes con este sistema operativo.
- Java SE para despliegues independientes en máquinas virtuales de Java.
- Java EE para despliegues en Servlet de Java.

2.2.2 SOAP

SOAP son las siglas de Simple Object Access Protocol, es un protocolo que permite la comunicación entre aplicaciones a través de mensajes a través de internet. Está basado en XML, es independiente de la plataforma y del lenguaje.

SOAP deriva de XML-RPC, un protocolo creado por David Winer en 1998, SOAP fue creado por Microsoft e IBM entre otros. Actualmente está bajo supervisión de la W3C.

Uno de los principales objetivos de SOAP es establecer un protocolo estándar de invocación de servicios remotos basados en protocolos estándares de Internet. HTTP para la transmisión y XML para la codificación de datos.

El protocolo HTTP es el protocolo utilizado para la conexión en Internet, utilizarlo para la transmisión en SOAP garantiza que cualquier cliente con un navegador estándar pueda conectarse con un servidor remoto. XML es un estándar de intercambio de datos, utilizarlo para la codificación de datos salva las incompatibilidades que poseen otros protocolos.

El segundo objetivo principal de SOAP es la independencia de plataforma, lenguaje de desarrollo e implementación.

Los mensajes SOAP son transmisiones unidireccionales desde un emisor a un receptor. Se suelen combinar mensajes para implementar patrones, como petición/respuesta. SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí.

Una aplicación SOAP debe procesar un mensaje siguiendo un orden de acciones:

1. Identificar las partes del mensaje SOAP dirigido a dicha aplicación.
2. Aceptar las partes obligatorias identificadas en el primer paso y procesarlas de forma adecuada, si no se procesan adecuadamente se deberá descartar el mensaje.
3. Si la aplicación SOAP no es el destino final del mensaje, quitar todas las partes identificadas en el primer paso antes de reenviar el mensaje.

En la ilustración 2.29 se puede observar la estructura del protocolo SOAP, en ella se pueden apreciar tres secciones diferenciadas.

- Envelope, o envoltura, es el elemento raíz del mensaje que sirve para describir el contenido y la forma de procesarlo.

- Header, o cabecera, en ella se incluye la información de la identificación del contenido.
- Body, o cuerpo, es el contenido del mensaje.

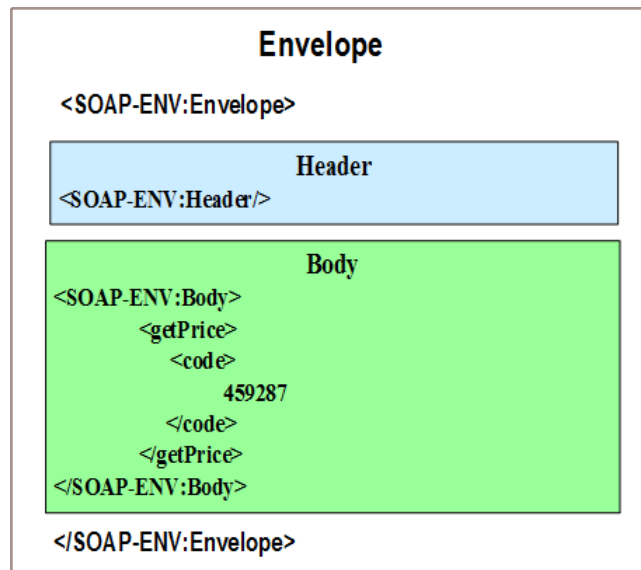


ILUSTRACIÓN 2.29: ESTRUCTURA SOAP

2.2.3 Comparativa

Para muchos diseñadores de servicios Web, SOAP es demasiado complicado, esta es la principal razón por la cual estos diseñadores pasan de utilizar SOAP a usar servicios Web basados en REST para mostrar grandes cantidades de datos, este es el caso de grandes empresas de Internet como Google, eBay, Yahoo! o Amazon, que decidieron diseñar servicios basados en REST.

Los servicios Web basados en REST se distinguen de los servicios basados en SOAP en que la interacción se produce a través de recursos, no de operaciones. Cada uno de estos recursos se identifica mediante una URL y corresponde con un elemento de información.

En la tabla 2.2 [18], se pueden ver las principales características de REST y SOAP.

TABLA 2.2: REST vs SOAP

	REST	SOAP
Características	<p>Las operaciones son definidas en los mensajes.</p> <p>Una dirección única para cada instancia del proceso.</p> <p>Cada objeto soporta las operaciones estándares definidas.</p> <p>Componentes débilmente acoplados.</p>	<p>Las operaciones son definidas como puertos WSDL.</p> <p>Dirección única para todas las operaciones.</p> <p>Múltiples instancias del proceso comparten la misma operación.</p> <p>Componentes fuertemente acoplados.</p>
Ventajas declaradas	<p>Bajo consumo de recursos.</p> <p>Las instancias del proceso son creadas explícitamente.</p> <p>El cliente no necesita información de enrutamiento a partir de la URI inicial.</p> <p>Los clientes pueden tener una interfaz escuchadora genérica para las notificaciones.</p> <p>Generalmente fácil de construir y adoptar.</p>	<p>Fácil de utilizar.</p> <p>La depuración es posible.</p> <p>Las operaciones complejas pueden ser escondidas detrás de una fachada.</p> <p>Envolver APIs existentes es sencillo.</p> <p>Incrementa la privacidad.</p> <p>Herramientas de desarrollo.</p>
Posibles desventajas	<p>Gran número de objetos.</p> <p>Manejar el espacio de nombres, URIs, puede ser complicado.</p> <p>La descripción sintáctica/semántica muy informal, está orientada al usuario.</p> <p>Pocas herramientas de desarrollo.</p>	<p>Los clientes necesitan saber las operaciones y su semántica antes del uso.</p> <p>Los clientes necesitan puertos dedicados para diferentes tipos de notificaciones.</p> <p>Las instancias del proceso son creadas implícitamente.</p>

En RES la interacción es dirigida por el usuario por medio de formulario, en SOAP hay un flujo de eventos orquestados. En REST son necesarias pocas operaciones con muchos recursos en cambio en SOAP son necesarias muchas operaciones con pocos recursos. REST posee un mecanismo consistente de nombrado de recursos, URI, SOAP no posee un mecanismo de nombrado. REST se centra en la escalabilidad y rendimiento a gran escala para sistemas distribuidos hipermedia en cambio SOAP se centra en el diseño de aplicaciones distribuidas.

REST es un protocolo síncrono que utiliza HTTP y XML. SOAP es un protocolo tanto síncrono como asíncrono, es independiente del transporte, como codificación utiliza

XML Schema. SOAP usa el lenguaje de Descripción de Servicios Web, WSDL, que permite describir con detalles el servicio Web, requiere tipado fuerte y se pueden construir automáticamente stubs, clientes. REST utiliza WADL, Web Application Description Language, no es necesario tipado fuerte, si ambos lados están de acuerdo con el contenido. En la parte de seguridad REST utiliza HTTPS, una comunicación punto a punto segura, SOAP utiliza WS-Security, una comunicación origen a destino segura.



ILUSTRACIÓN 2.30: SERVICIOS WEB

Un servicio REST es recomendable usarlo cuando el servicio Web no tiene estado, si la interacción es capaz de sobrevivir a un reinicio del sistema. Cuando una infraestructura de caching puede mejorar el rendimiento, si los datos que el servicio Web devuelve no son dinámicamente generados y pueden ser cacheados la infraestructura de caching que los servidores Web y los intermediarios proporcionan pueden incrementar el rendimiento. Cuando el ancho de banda es importante y necesita ser limitado, REST es particularmente útil en dispositivos con escasos recursos como PDAs o teléfonos móviles, donde la sobrecarga de las cabeceras y capas adicionales de los elementos SOAP debe ser restringida.

Un servicio basado en SOAP es adecuado cuando se establece un contrato formal para la descripción de la interfaz que el servicio ofrece. SOAP también es adecuado cuando la arquitectura debe abordar requerimientos complejos no funcionales. Cuando la arquitectura necesita manejar procesado asíncrono e invocación, en estos casos, la infraestructura proporcionada por estándares como WSRM y APIs como JAX-WS junto con la asincronía por el lado del cliente permitirán el soporte de estas características.

2.3 Aplicaciones de monitorización

En esta sección se expondrán las características de las principales de algunas aplicaciones de monitorización, aplicaciones con fines similares a la aplicación desarrollada en este proyecto.

2.3.1 SCADA

SCADA son las siglas de “Supervisory Control And Data Acquisition”, Adquisición de datos y control de supervisión, se trata de aplicaciones de software diseñadas para controlar y supervisar procesos a distancia basadas en la adquisición de datos de los procesos remotos [19].

Este software se ejecuta en la mayoría de los casos en ordenadores que efectúen tareas de supervisión, gestión de alarmas, tratamiento de datos y control de procesos. La comunicación se realiza mediante buses o redes LAN. La ejecución es normalmente en tiempo real.

Un sistema SCADA debe ofrecer las siguientes funcionalidades [20]:

Supervisión remota de instalaciones y equipos: Permite al operador conocer el estado de desempeño de las instalaciones y los equipos alojados en la planta, lo que permite dirigir las tareas de mantenimiento y estadísticas de errores.

Control remoto de instalaciones y equipos: Mediante el sistema se puede activar o desactivar los equipos remotamente tanto manual como automáticamente.

Generación de reportes: Posibilidad de crear paneles de alarma con registro de incidencias, exige la presencia del operador para reconocer la parada o situación de alarma.

Generación de datos históricos: Almacenamiento de las señales, se debe tener también la opción de volcado en hojas de cálculos.

Ejecución de programas, que modifiquen la ley de control o anular las tareas asociadas bajo ciertas condiciones.

Posibilidad de programación numérica que permita realizar cálculos aritméticos de elevada resolución sobre la CPU del ordenador.

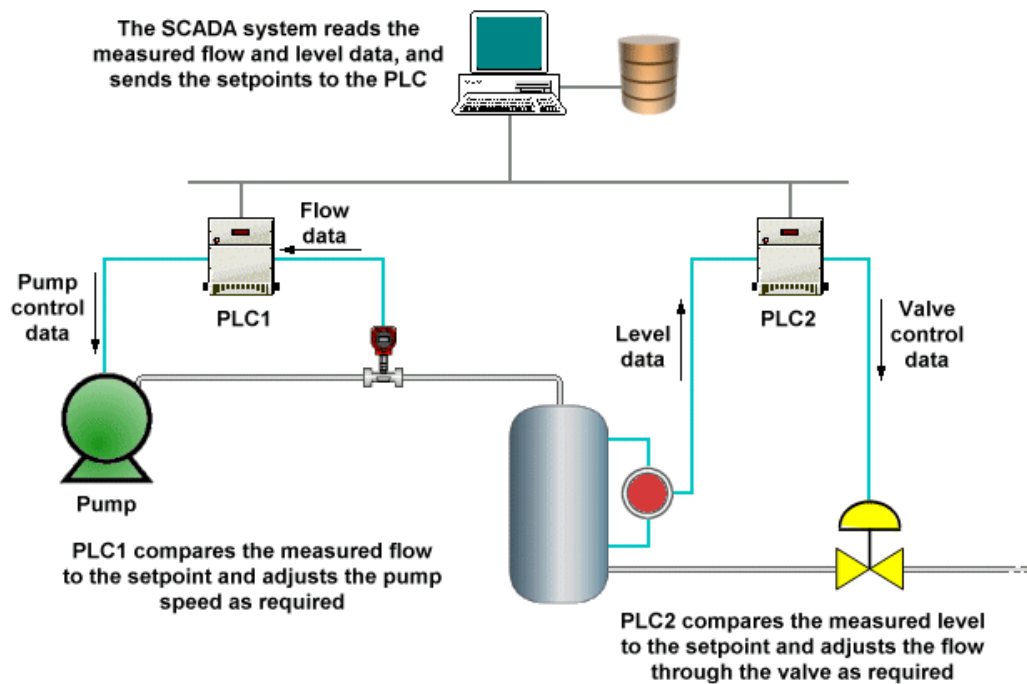


ILUSTRACIÓN 2.31: UN SISTEMA SCADA TÍPICO

SCADA normalmente incluye los siguientes subsistemas:

Interfaz hombre-máquina (HMI): es la interfaz entre el hardware de control de procesos y un operador humano.

Sistema de control: por o general un sistema informático que adquiere los datos del proceso y los comandos de las cuestiones de control para el proceso.

Unidades terminales remotas (RTUs): Son los elementos que envían algún tipo de información a la unidad central.

Controlador lógico programable (PLCs): Son las unidades que toman las señales y las envían a las estaciones remotas utilizando un determinado protocolo.

Infraestructura de comunicación: es el elemento que conecta los sistemas de supervisión con las unidades terminales remotas y los controladores lógicos programables.

3 Marco regulador

La aplicación que se pretende desarrollar en este proyecto puede verse afectada por la Ley orgánica de protección de datos. En este capítulo se expondrán los detalles principales de dicha Ley para conocer si realmente puede afectar o no al sistema que concierne a este proyecto.

La Ley orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal es una ley española que tiene como objetivo garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas y en especial su honor, intimidad y privacidad personal y familiar.

La Ley orgánica de Protección de Datos de Carácter Personal establece las obligaciones que los responsables de los ficheros o tratamientos y los encargados de los tratamientos, tanto de organismos públicos como privados, han de cumplir para garantizar el derecho a la protección de datos de carácter personal [21].

La Ley orgánica de Protección de Datos de Carácter Personal afecta a cualquier profesional liberal, organización pública o privada o empresa que guarde algún dato de carácter personal, como nombre o apellidos, de cualquier persona, sea empleado, cliente, proveedor; Independientemente del formato.

Para que la aplicación que se pretende desarrollar no tenga problemas con esta ley se tratarán la información sensible que se almacene en las bases de datos de la aplicación. Para ello se almacenarán cifrados los datos sensibles de los usuarios como las contraseñas de acceso de los mismos.

4 Análisis

En esta sección se expondrá un análisis detallado del sistema con el objetivo de facilitar los posteriores diseños y desarrollos del proyecto que serán expuestos en capítulos posteriores de este documento.

4.1 Metodología

El uso de una metodología como ESA o Métrica v3 supondría invertir demasiado tiempo para desarrollar un proyecto con un tiempo tan limitado. Por tanto se ha decantado para el desarrollo de este proyecto por el uso de una metodología ágil.

La metodología elegida para el desarrollo ha sido eXtreme Programming o XP, una metodología formulada por Kent Beck [22]. Es considerada una de las metodologías ágiles más utilizadas.

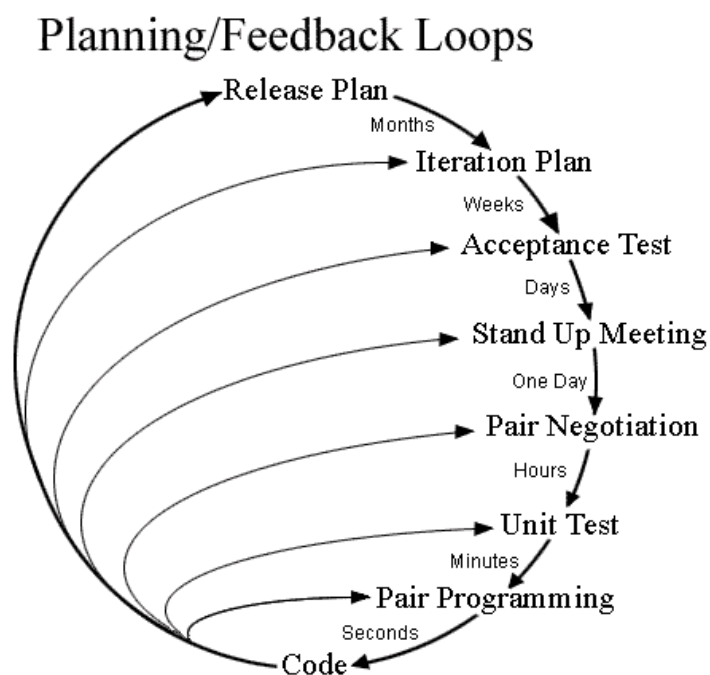


ILUSTRACIÓN 4.1: eXtreme PROGRAMMING

Las características principales de la eXtreme Programming o programación extrema son:

- Desarrollo interactivo e incremental.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.

- Programación en pareja: En esta metodología se recomienda que las tareas de desarrollo se lleven a cabo en parejas. Escribir un código en pareja aumenta la calidad del mismo ya que este es comentado y discutido mientras se escribe.
- Frecuente integración del equipo de programación con el cliente o usuario. Es recomendable que un representante del cliente trabaje junto con el equipo de desarrollo.
- Corrección de errores antes de añadir nuevas funcionalidades.
- Refactorización del código, rescribir ciertas partes del código aumenta su legibilidad y mantenimiento.
- Simplicidad del código, cuanto más sencillo sea el código más fácil será de modificar y añadir nuevas funcionalidades más adelante.

4.2 Análisis del proyecto

En este apartado se expondrán los detalles del análisis del proyecto. En primer lugar se enumerarán los casos de uso del proyecto. De los casos de uso se extraerán los requisitos de usuario que serán expuestos a continuación. A continuación se expondrán los requisitos de software extraídos de los requisitos de usuario y se analizarán las relaciones entre los primeros y los segundos. Finalmente se analizará la estructura de las bases de datos necesarias para el desarrollo del proyecto.

4.2.1 Modelo de casos de uso

En este apartado se analizará el modelo de casos de uso, se comenzará enumerando los actores participantes, se proseguirá analizando las características de los casos de uso del sistema y se finalizará con una representación gráfica del modelo resultante.

4.2.1.1 Actores

Los actores son aquellas personas que interactúan con el sistema.

4.2.1.1.1 Usuario

El actor usuario es el propietario del dispositivo móvil en el que se instala y ejecuta la aplicación de monitorización. Este actor interactúa tanto con el dispositivo móvil como con el servidor. A diferencia del actor administrador en el servidor sólo tendrá acceso a los datos de monitorización de su propio dispositivo monitorizado.

4.2.1.1.2 Administrador

El actor administrador solamente interactúa con el servidor web. Es capaz de ver la monitorización de todos los dispositivos móviles que se encuentran conectados a la aplicación.

4.2.1.2 Casos de uso

Los casos de uso que se enumerarán a continuación cuentan con los siguientes atributos.

- **Nombre:** Identificación del caso de uso.
- **Descripción:** Descripción de la funcionalidad del caso de uso.
- **Actores:** Actores relacionados con el caso de uso.
- **Precondiciones:** Acciones o hechos que se han de cumplir para que el caso de uso se pueda iniciar.
- **Flujo normal:** serie de eventos que se desarrollan si el caso de uso se ejecuta sin problemas.
- **Flujo Alternativo:** serie de eventos que se desarrollan si se produce algún fallo al ejecutar el caso de uso.
- **Postcondiciones:** Acciones o hechos que se cumplirán si el flujo de eventos normal se ha ejecutado correctamente.

TABLA 4.1: CASO DE USO 1: DESCARGAR APLICACIÓN MÓVIL

Nombre	Descargar aplicación móvil
Descripción	Se descarga el archivo de instalación en el dispositivo.
Actores	Usuario.
Precondiciones	El usuario ha accedido a la página web.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario accede a la página web. 2. El usuario pulsa el enlace “descargar app”. 3. Se descarga la aplicación en el dispositivo.
Flujo Alternativo	-
Postcondiciones	El archivo de instalación .apk de la aplicación queda guardado en el dispositivo.

TABLA 4.2: CASO DE USO 2: REGISTRARSE EN LA APLICACIÓN

Nombre	Registrarse en la aplicación
Descripción	En usuario se registra en la aplicación.
Actores	Usuario.
Precondiciones	La aplicación debe estar instalada en el teléfono móvil y en primer plano.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario introduce en el formulario los datos solicitados, el correo electrónico, la contraseña y el tiempo de periodo. 2. El usuario pulsa el botón enviar.
Flujo Alternativo	<ol style="list-style-type: none"> 3. El sistema muestra un aviso informando de un error.
Postcondiciones	El usuario es redirigido al menú principal de la aplicación.

TABLA 4.3: CASO DE USO 3: IDENTIFICARSE EN LA APLICACIÓN

Nombre	Identificarse en la aplicación
Descripción	El usuario introduce los datos introducidos inicialmente al registrarse para acceder a la aplicación.
Actores	Usuario.
Precondiciones	La aplicación debe estar instalada en el teléfono móvil y en primer plano.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario introduce en el formulario los datos solicitados, el correo electrónico, la contraseña y el tiempo de periodo. 2. El usuario pulsa el botón aceptar.
Flujo Alternativo	<ol style="list-style-type: none"> 3. El sistema muestra un aviso informando de un error.
Postcondiciones	El usuario es redirigido al menú principal de la aplicación.

TABLA 4.4: CASO DE USO 4: VER PERFIL

Nombre	Ver perfil
Descripción	Muestra los datos de configuración, el correo electrónico y el tiempo de periodo, al usuario.
Actores	Usuario.
Precondiciones	La aplicación debe estar instalada y abierta, el usuario debe estar identificado en la aplicación.
Flujo Normal	1. El usuario pulsa el icono perfil del menú principal.
Flujo Alternativo	-
Postcondiciones	La aplicación es dirigida a la pantalla perfil.

TABLA 4.5: CASO DE USO 5: ACTIVAR/DESACTIVAR MONITORIZACIÓN

Nombre	Activar/desactivar monitorización
Descripción	Se activa o se desactiva la monitorización del dispositivo.
Actores	Usuario.
Precondiciones	La aplicación debe estar instalada y abierta, el usuario debe estar identificado en la aplicación.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario pulsa el icono opciones del menú principal. 2. El usuario pulsa el botón de monitorización del menú de opciones.
Flujo Alternativo	-
Postcondiciones	Se muestra un mensaje avisando de que la monitorización se ha activado o desactivado según el caso.

TABLA 4.6: CASO DE USO 6: CAMBIAR PERIODO

Nombre	Cambiar periodo
Descripción	Cambiar el valor del periodo de monitorización.
Actores	Usuario.
Precondiciones	La aplicación debe estar instalada y abierta, el usuario debe estar identificado en la aplicación.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario pulsa el icono opciones del menú principal. 2. El usuario pulsa el botón de monitorización del menú de opciones.
Flujo Alternativo	-
Postcondiciones	Se muestra un mensaje avisando del cambio correcto del periodo.

TABLA 4.7: CASO DE USO 7: MONITORIZAR DESDE TERMINAL

Nombre	Monitorizar desde terminal
Descripción	Muestra una lista de los sensores monitorizados con los valores capturados cada cierto periodo seleccionado.
Actores	Usuario
Precondiciones	La aplicación debe estar instalada y abierta, el usuario debe estar identificado en la aplicación.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario pulsa el monitorizar opciones del menú principal.
Flujo Alternativo	-
Postcondiciones	El usuario es redirigido a la pantalla de monitorización de la aplicación.

TABLA 4.8: CASO DE USO 8: IDENTIFICARSE EN EL SERVIDOR WEB

Nombre	Identificarse en el servidor web
Descripción	El usuario introduce los datos introducidos inicialmente al registrarse para acceder a la web.
Actores	Usuario, Administrador.
Precondiciones	El usuario se encuentra en la página de identificación de la web.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario introduce en el formulario los datos solicitados, el correo electrónico, la contraseña y el tiempo de periodo. 2. El usuario pulsa el botón aceptar.
Flujo Alternativo	<ol style="list-style-type: none"> 3. El sistema muestra un aviso informando de un error.
Postcondiciones	El usuario es redirigido al panel principal de monitorización de la web.

TABLA 4.9: CASO DE USO 9: MONITORIZACIÓN DISTRIBUIDA

Nombre	Monitorización distribuida
Descripción	El usuario puede ver los datos del dispositivo monitorizado, lista de sensores y datos obtenidos, desde la aplicación web.
Actores	Usuario, Administrador.
Precondiciones	El usuario debe haber accedido a la página web
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario se identifica en la página web con su correo y contraseña.
Flujo Alternativo	-
Postcondiciones	El usuario es redirigido al panel principal de la página.

TABLA 4.10: CASO DE USO 10: VER DETALLES DE MONITORIZACIÓN

Nombre	Ver detalles de monitorización
Descripción	El usuario puede ver con detalles la monitorización de un dispositivo.
Actores	Usuario, Administrador.
Precondiciones	El usuario se ha identificado en la web y se encuentra en el panel principal de monitorización.
Flujo Normal	1. El usuario pulsa sobre la fila de identificación del dispositivo del cual quiere ver los detalles.
Flujo Alternativo	-
Postcondiciones	El usuario es redirigido al panel de monitorización detallada de la web.

TABLA 4.11: CASO DE USO 11: CONSULTAR HISTÓRICO

Nombre	Consultar histórico
Descripción	El usuario puede consultar el histórico de la monitorización de los dispositivos monitorizados.
Actores	Usuario, Administrador.
Precondiciones	El usuario se ha identificado en la web y se encuentra en el panel principal de monitorización.
Flujo Normal	1. El usuario pulsa sobre el enlace histórico. 2. El usuario rellena el formulario seleccionando el usuario, el dispositivo las fechas que quiere monitorizar.
Flujo Alternativo	-
Postcondiciones	La web muestra una tabla con los datos de la monitorización comprendidos entre las fechas seleccionadas.

TABLA 4.12: CASO DE USO 12: SELECCIONAR IDIOMA

Nombre	Seleccionar idioma
Descripción	El usuario podrá cambiar el idioma en el que se muestra la página web.
Actores	Usuario, Administrador.
Precondiciones	El usuario ha accedido a la página web.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la casilla de selección de idioma situado al pie de página en la página principal o en la cabecera en las demás. 2. El usuario selecciona el idioma de la lista desplegada.
Flujo Alternativo	-
Postcondiciones	La web muestra la misma pantalla en el idioma seleccionado

TABLA 4.13: CASO DE USO 13: CAMBIAR CONTRASEÑA

Nombre	Cambiar contraseña
Descripción	El usuario podrá cambiar la contraseña de acceso desde la aplicación móvil.
Actores	Usuario.
Precondiciones	La aplicación debe estar instalada y abierta, el usuario debe estar identificado en la aplicación.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el icono opciones del menú principal. 2. El usuario introduce en la casilla "actual contraseña" la contraseña de la aplicación. 3. El usuario introduce en la casilla "nueva contraseña" la nueva contraseña que desee. 4. El usuario pulsa el botón enviar situado debajo de las casillas anteriores.
Flujo Alternativo	<ol style="list-style-type: none"> 5. La aplicación muestra un mensaje avisando de que se ha producido un error y no se ha llevado a cabo el cambio de contraseña.
Postcondiciones	La aplicación muestra un mensaje avisando de que el cambio se ha efectuado correctamente.

4.2.1.3 Modelo UML

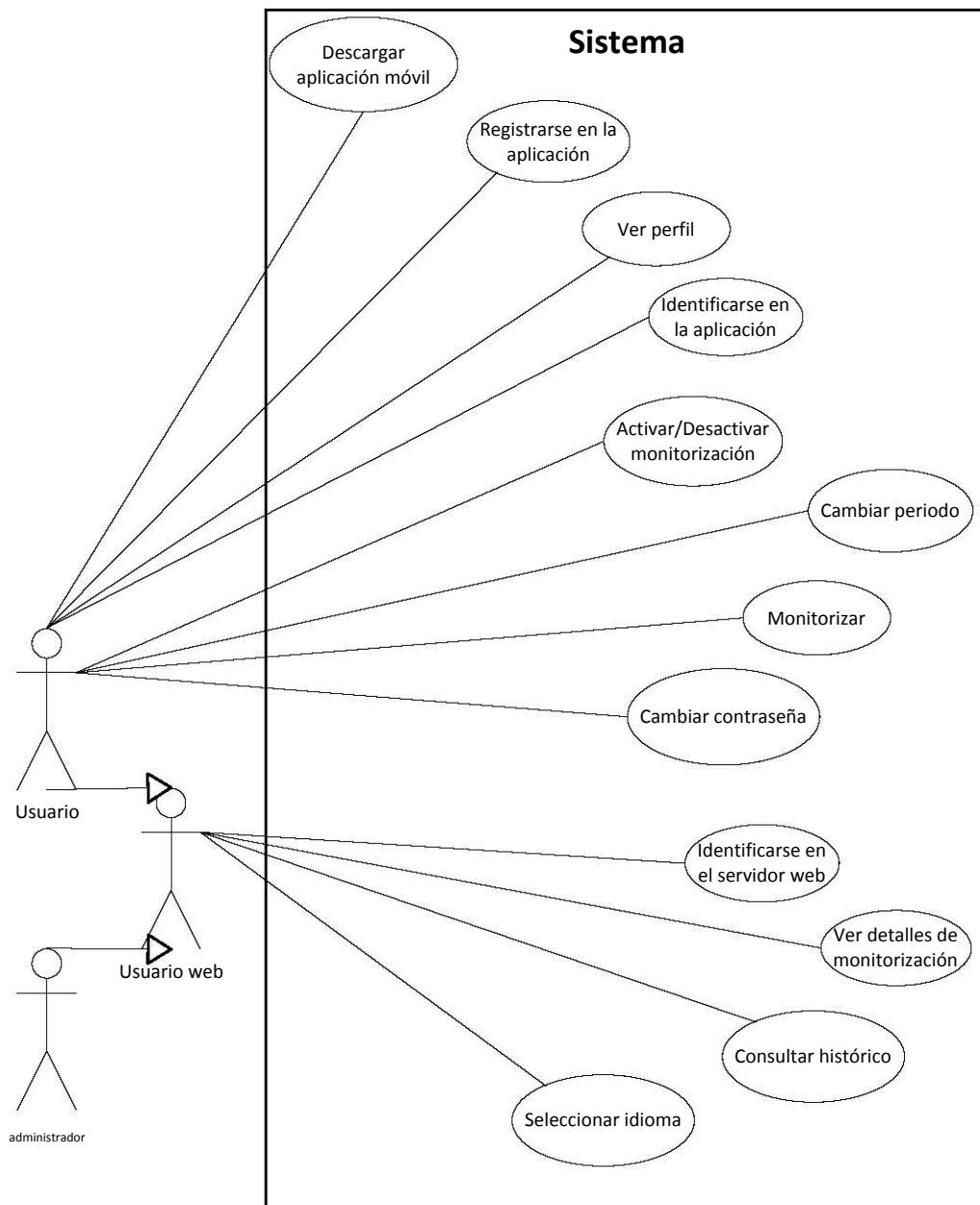


ILUSTRACIÓN 4.2: MODELO DE CASOS DE USO

El usuario es aquella persona que utiliza la aplicación desde su dispositivo móvil, si esta persona pasa a usar la aplicación web adquiere los casos de uso del Usuario Web. El administrador sólo existe como Usuario web, se distingue del usuario normal en la cantidad de información a la que puede acceder en la página.

4.2.2 Análisis de requisitos

Los requisitos de usuario son extraídos de los casos de uso expuestos en el apartado anterior. A continuación se expondrán todos los requisitos de usuario extraídos y seguidamente se relacionarán con los casos de uso ya explicados.

Todos los requisitos expuestos en los siguientes apartados cuentan con los siguientes atributos.

- **Nombre:** Identificación del caso de uso.
- **Descripción:** Descripción de la funcionalidad del requisito.
- **Estabilidad:** Establece si el requisito podría ser modificado con el tiempo.
- **Necesidad:** Establece el nivel de necesidad del requisito.
- **Prioridad:** Establece el nivel de prioridad del requisito.
- **Verificabilidad:** Establece como es verificable la funcionalidad del requisito.
- **Claridad:** Establece como de preciso es el requisito.
- **Fuente:** Marca de donde proviene el requisito.

4.2.2.1 Requisitos de usuario

En este apartado se listarán los requisitos de usuario del sistema, estarán divididos en dos categorías, los requisitos de capacidad y los requisitos de restricción.

4.2.2.1.1 Requisitos de capacidad

En este punto se listará los requisitos de capacidad del proyecto.

TABLA 4.14: REQUISITO DE USUARIO 1: REGISTRO

RU01	Registro				
Descripción	Los usuarios pueden registrarse a través de la aplicación móvil.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Muy Alta
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.15: REQUISITO DE USUARIO 2: IDENTIFICACIÓN EN LA APLICACIÓN MÓVIL

RU02	Identificación en la aplicación móvil				
Descripción	Los usuarios pueden identificarse para acceder a la aplicación móvil en sus dispositivos.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Muy Alta
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.16: REQUISITO DE USUARIO 3: IDENTIFICACIÓN EN LA APLICACIÓN WEB

RU03	Identificación en la aplicación web				
Descripción	Los usuarios pueden identificarse para acceder a la página web.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Muy Alta
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.17: REQUISITO DE USUARIO 4: DESCARGA APLICACIÓN

RU04	Descarga aplicación				
Descripción	Cualquier usuario puede descargar el archivo instalador de la aplicación web desde la página web.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Alta
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.18: REQUISITO DE USUARIO 5: VER DATOS DE USUARIO

RU05	Ver datos de usuario				
Descripción	Los usuarios podrán ver los datos de su perfil en la aplicación móvil.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Media
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.19: REQUISITO DE USUARIO 6: CAMBIAR ESTADO DE LA MONITORIZACIÓN

RU06 Cambiar estado de la monitorización					
Descripción	Los usuarios podrán activar o desactivar la monitorización de los sensores del dispositivo desde el mismo terminal.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Media
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.20: REQUISITO DE USUARIO 7: CAMBIAR PERIODO DE LA MONITORIZACIÓN

RU07 Cambiar periodo de la monitorización					
Descripción	Los usuarios podrán cambiar el tiempo de periodo de la monitorización del dispositivo desde la aplicación móvil.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Media
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.21: REQUISITO DE USUARIO 8: MONITORIZAR DISPOSITIVO

RU08 Monitorizar dispositivo					
Descripción	Los usuarios podrán ver la lista de sensores y los datos monitorizados de estos desde la aplicación móvil.				
Estabilidad	Estable	Necesidad	Deseable	Prioridad	Media
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.22: REQUISITO DE USUARIO 9: MONITORIZAR DISTRIBUIDA DEL DISPOSITIVO

RU09 Monitorizar distribuida del dispositivo					
Descripción	Los usuarios podrán ver la lista de sensores y los datos monitorizados de estos desde la página web.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Alta
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.23: REQUISITO DE USUARIO 10: MONITORIZACIÓN DETALLADA

RU10	Monitorización detallada				
Descripción	Los usuarios podrán ver detalladamente la monitorización de un dispositivo.				
Estabilidad	Estable	Necesidad	Deseable	Prioridad	Media
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.24: REQUISITO DE USUARIO 11: VER HISTÓRICO

RU11	Ver histórico				
Descripción	<p>Los usuario pueden consultar el histórico de la monitorización de un dispositivo mediante un formulario en el que tendrá que introducir:</p> <ul style="list-style-type: none"> - usuario - dispositivo - fecha de inicio - fecha de fin 				
Estabilidad	Estable	Necesidad	Deseable	Prioridad	Baja
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.25: REQUISITO DE USUARIO 12: SELECCIONAR IDIOMA

RU12	Seleccionar idioma				
Descripción	El usuario podrá cambiar el idioma de la página web.				
Estabilidad	Estable	Necesidad	Deseable	Prioridad	Baja
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.26: REQUISITO DE USUARIO 13: CAMBIAR CONTRASEÑA

RU13	Cambiar contraseña				
Descripción	El usuario podrá cambiar su contraseña de acceso desde la aplicación móvil				
Estabilidad	Estable	Necesidad	Deseable	Prioridad	Baja
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

4.2.2.1.2 Requisitos de restricción

En este punto se listará los requisitos de restricción del proyecto.

TABLA 4.27: REQUISITO DE RESTRICCIÓN 1: REGISTRO EN APLICACIÓN

RR01	Registro en aplicación				
Descripción	Sólo los usuarios registrados e identificados podrán acceder a las funcionalidades de la aplicación móvil.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Alta
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.28: REQUISITO DE RESTRICCIÓN 2: IDENTIFICACIÓN EN EL SERVIDOR

RR02	Identificación en el servidor				
Descripción	Sólo los usuarios correctamente identificados pueden acceder a la página web en el servidor.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Alta
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.29: REQUISITO DE RESTRICCIÓN 3: REGISTRO

RR03	Registro				
Descripción	Los usuarios no administradores sólo podrán ver en la web los datos de monitorización de los dispositivos asociados a su cuenta.				
Estabilidad	Estable	Necesidad	Esencial	Prioridad	Alta
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.30: REQUISITO DE RESTRICCIÓN 4: IDIOMAS

RR04	Idiomas				
Descripción	Los lenguajes disponibles en la web serán: <ul style="list-style-type: none"> - Español - Inglés 				
Estabilidad	No Estable	Necesidad	Deseable	Prioridad	Baja
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.31: REQUISITO DE RESTRICCIÓN 5: CAMBIO DE CONTRASEÑA

RR05	Cambio de contraseña				
Descripción	Para cambiar la contraseña en la aplicación móvil el usuario deberá introducir primero su antigua contraseña.				
Estabilidad	No Estable	Necesidad	Deseable	Prioridad	Baja
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

TABLA 4.32: REQUISITO DE RESTRICCIÓN 6: VALOR MÍNIMO PERIODO

RR06	Valor mínimo del periodo				
Descripción	El valor mínimo admitido por el periodo es 1.				
Estabilidad	No Estable	Necesidad	Esencial	Prioridad	Media
Verificabilidad	Alta	Claridad	Alta	Fuente	Cliente

4.2.2.2 Trazabilidad: casos de uso – requisitos de usuario

En la tabla 3.32 se puede apreciar de qué caso de uso se han extraído cada requisito de usuario anteriormente expuesto.

TABLA 4.33: TRAZABILIDAD: CASOS DE USO – REQUISITOS DE USUARIO

	CU01	CU02	CU03	CU04	CU05	CU06	CU07	CU08	CU09	CU10	CU11	CU12	CU13
RU01		X											
RU02			X										
RU03								X					
RU04	X												
RU05				X									
RU06					X								
RU07						X							
RU08							X						
RU09									X				
RU10									X	X			
RU11									X		X		
RU12												X	
RU13													X
RR01		X	X										
RR02								X					
RR03									X	X	X		
RR04												X	
RR05													X
RR06	X						X						

4.2.2.3 Requisitos de software

Los requisitos de software son el resultado del análisis de los requisitos de usuario expuestos en la sección anterior. En este apartado se listarán los requisitos de software extraídos, sus relaciones entre ellos y entre los requisitos de usuario.

Los requisitos de software se clasificarán en dos apartados, los requisitos funcionales y los requisitos no funcionales.

Todos los requisitos contarán con los siguientes atributos.

- **Identificador:** Identificación del caso de uso.
- **Descripción:** Descripción de la funcionalidad del requisito.
- **Tipo:** Establece el tipo del requisito, funcional o no funcional.
- **Necesidad:** Establece el nivel de necesidad del requisito.
- **Prioridad:** Establece el nivel de prioridad del requisito.
- **Verificabilidad:** Establece como es de verificable la funcionalidad del requisito.

4.2.2.3.1 Requisitos funcionales

En este punto se listarán los requisitos funcionales del proyecto.

TABLA 4.34: REQUISITO DE SOFTWARE 1: REGISTRO

RS01			
Descripción	El software deberá permitir al usuario registrarse en la aplicación a través de la aplicación móvil.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.35: REQUISITO DE SOFTWARE 2: IDENTIFICACIÓN APLICACIÓN

RS02			
Descripción	El software debe permitir al usuario identificarse en la aplicación móvil al abrirla.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.36: REQUISITO DE SOFTWARE 3: IDENTIFICACIÓN WEB

RS03			
Descripción	El software debe permitir al usuario identificarse en la página web para acceder a sus funcionalidades.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.37: REQUISITO DE SOFTWARE 4: DESCARGA APLICACIÓN

RS04			
Descripción	El software debe permitir la descarga del ejecutable de instalación de la aplicación móvil a todo usuario desde la página web		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Media	Verificabilidad	Alta

TABLA 4.38: REQUISITO DE SOFTWARE 5: DATOS DE PERFIL

RS05			
Descripción	El software debe permitir al usuario ver los datos de su perfil en la aplicación móvil.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Media	Verificabilidad	Alta

TABLA 4.39: REQUISITO DE SOFTWARE 6: MONITORIZACIÓN

RS06			
Descripción	El software debe monitorizar los sensores del dispositivo móvil en el que está instalado cada cierto tiempo introducido por el usuario, tomar los valores de los sensores y enviarlos al servidor.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.40: REQUISITO DE SOFTWARE 7: MONITORIZACIÓN SIN CONEXIÓN

RS07			
Descripción	En caso de que el dispositivo no disponga de conexión a internet, los datos monitorizados se almacenarán en la base de datos del dispositivo y se enviarán cuando se recupere la conexión.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.41: REQUISITO DE SOFTWARE 8: MODIFICACIÓN DE LA MONITORIZACIÓN

RS08			
Descripción	El software debe dar la opción al usuario de activar o desactivar la monitorización del dispositivo desde el mismo terminal.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.42: REQUISITO DE SOFTWARE 9: INTRODUCIR PERIODO

RS09			
Descripción	El software debe dar la opción al usuario de introducir el periodo de monitorización del dispositivo en al realizar el registro.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.43: REQUISITO DE SOFTWARE 10: CAMBIAR PERIODO

RS10			
Descripción	El software debe dar la opción al usuario de cambiar el periodo de la monitorización del dispositivo desde el mismo terminal.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.44: REQUISITO DE SOFTWARE 11: MONITORIZACIÓN DESDE DISPOSITIVO

RS11			
Descripción	El software debe dar la opción al usuario de ver la lista de sensores y los datos monitorizados desde la aplicación móvil.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Media	Verificabilidad	Alta

TABLA 4.45: REQUISITO DE SOFTWARE 12: MONITORIZACIÓN DISTRIBUIDA

RS12			
Descripción	El software debe dar la opción al usuario de ver la lista de sensores y los datos monitorizados desde la aplicación web.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.46: REQUISITO DE SOFTWARE 13: MONITORIZACIÓN DETALLADA

RS13			
Descripción	El software debe dar la opción al usuario de ver con más detalle los datos de la monitorización de los dispositivos.		
Tipo	Funcional	Necesidad	Deseable
Prioridad	Media	Verificabilidad	Alta

TABLA 4.47: REQUISITO DE SOFTWARE 14: HISTÓRICO

RS14			
Descripción	<p>El software debe dar la opción al usuario de ver el histórico de la monitorización de un dispositivo mediante un formulario en la página web introduciendo los siguientes datos:</p> <ul style="list-style-type: none"> - usuario - dispositivo - fecha de inicio - fecha de fin 		
Tipo	Funcional	Necesidad	Deseable
Prioridad	Media	Verificabilidad	Alta

TABLA 4.48: REQUISITO DE SOFTWARE 15: AUTENTIFICACIÓN

RS15			
Descripción	El software deberá comprobar la correcta autenticación del usuario antes de mostrarles la información en la aplicación móvil.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.49: REQUISITO DE SOFTWARE 16: AUTENTIFICACIÓN WEB

RS16			
Descripción	El software deberá comprobar la correcta autenticación de los usuarios antes de mostrarles la información en la página web.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.50: REQUISITO DE SOFTWARE 17: DISPOSITIVOS ASOCIADOS

RS17			
Descripción	El software sólo deberá permitir a los usuarios ver los datos monitorizados de los dispositivos asociados a su cuenta.		
Tipo	Funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Alta

TABLA 4.51: REQUISITO DE SOFTWARE 18: CAMBIAR IDIOMA

RS18			
Descripción	El software sólo deberá permitir a los usuarios cambiar el idioma de la página web.		
Tipo	Funcional	Necesidad	Deseable
Prioridad	Baja	Verificabilidad	Alta

TABLA 4.52: REQUISITO DE SOFTWARE 19: IDIOMAS

RS19			
Descripción	Los idiomas que los usuarios podrán seleccionar serán: <ul style="list-style-type: none"> - Español - Inglés 		
Tipo	Funcional	Necesidad	Deseable
Prioridad	Baja	Verificabilidad	Alta

TABLA 4.53: REQUISITO DE SOFTWARE 20: CAMBIAR CONTRASEÑA

RS20			
Descripción	El software deberá permitir al usuario cambiar la contraseña desde la aplicación móvil.		
Tipo	Funcional	Necesidad	Deseable
Prioridad	Baja	Verificabilidad	Alta

TABLA 4.54: REQUISITO DE SOFTWARE 21: CAMBIO DE CONTRASEÑA

RS21			
Descripción	El software deberá solicitar al usuario su antigua contraseña y verificarla antes de dejarle cambiarla.		
Tipo	Funcional	Necesidad	Deseable
Prioridad	Baja	Verificabilidad	Alta

4.2.2.3.2 Requisitos no funcionales

En este apartado se listarán los requisitos no funcionales del proyecto.

TABLA 4.55: REQUISITO DE SOFTWARE 22: LENGUAJES DE PROGRAMACIÓN

RS22			
Descripción	El servidor web será desarrollado en los lenguajes HTML, CSS, JavaScript y PHP.		
Tipo	No funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Media

TABLA 4.56: REQUISITO DE SOFTWARE 23: LENGUAJE APLICACIÓN

RS23			
Descripción	La aplicación móvil será desarrollada en el lenguaje de programación Java.		
Tipo	No funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Media

TABLA 4.57: REQUISITO DE SOFTWARE 24: VERSIÓN SISTEMA OPERATIVO

RS24			
Descripción	La versión mínima del sistema operativo necesaria para instalar la aplicación móvil será Android 2.2.		
Tipo	No funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Media

TABLA 4.58: REQUISITO DE SOFTWARE 25: CONTRASEÑAS

RS25			
Descripción	La contraseña no se guardará en las bases de datos ni se enviara en claro, se guardará y se enviará la función hash de esta sumada a una cadena aleatoria.		
Tipo	No funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Media

TABLA 4.59: REQUISITO DE SOFTWARE 26: CONEXIÓN SEGURA

RS26			
Descripción	La conexión a la página web se realizará mediante SSL por la seguridad de los usuarios.		
Tipo	No funcional	Necesidad	Esencial
Prioridad	Alta	Verificabilidad	Media

4.2.2.4 Correspondencia

En la tabla 3.59 se puede apreciar la relación entre los requisitos de usuario expuestos en la sección anterior y los requisitos de software expuestos anteriormente, y una breve descripción de la funcionalidad del requisito de usuario.

TABLA 4.60: TABLA DE CORRESPONDENCIA DE REQUISITOS

Requisito de Usuario	Requisitos de Software	Descripción
RU01	RS01	Sólo los usuarios registrados pueden acceder a la aplicación.
RU02	RS02	Los usuarios deben identificarse para acceder a la aplicación móvil.
RU03	RS03	Los usuarios deben identificarse para acceder a la página web.
RU04	RS04	Los usuarios pueden descargarse el instalador de la aplicación.
RU05	RS05	Los usuarios pueden ver los datos de su perfil en la aplicación móvil.
RU06	RS06, RS07, RS08	La aplicación debe monitorizar los valores de los sensores del dispositivo.
RU07	RS09, RS10, RS06	Los usuarios podrán cambiar el valor del periodo de monitorización del dispositivo.
RU08	RS11	Los usuarios podrán ver los datos de la monitorización desde la aplicación móvil.
RU09	RS12	Los usuarios podrán ver los datos de la monitorización desde la página web.
RU10	RS13	Los usuarios podrán ver los datos de la monitorización

		con detalles desde la aplicación móvil.
RU11	RS14	Los usuarios podrán ver el histórico de la monitorización de un determinado dispositivo.
RU12	RS18, RS19	El usuario podrá cambiar el idioma de la página web
RU13	RS20, RS21	El usuario podrá cambiar su contraseña de acceso.
RR01	RS15	Los usuarios deberán estar identificados para acceder a la aplicación móvil
RR02	RS16	Los usuarios deberán estar identificados para acceder a la página web.
RR03	RS17	Los usuarios sólo podrán ver los datos de los dispositivos asociados a su cuenta.
RR04	RS19	Los idiomas seleccionables son: español e inglés.
RR05	RS21	Para cambiar la contraseña el usuario deberá introducir primero su antigua contraseña.

4.2.2.5 Matriz de consistencia

En la tabla 4.60 se puede apreciar la relación entre los requisitos de software analizados en los apartados anteriores.

TABLA 4.61: MATRIZ DE CONSISTENCIA

RS	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
01		+	+		+				+								+						+	+	+	
02	+							+		+	+				+					+			+	+	+	
03	+											+	+	+		+	+					+				+
04																						+				+
05	+																									
06							+	+	+	+	+	+	+	+			+						+	+		
07						+								+									+	+		
08		+				+																				
09	+					+				+	+	+	+										+	+		
10		+				+			+		+	+	+										+	+		
11		+				+			+	+													+	+		
12			+			+			+	+			+	+									+	+		
13			+			+			+	+		+										+				+
14			+			+	+					+										+				+
15		+																								
16			+														+					+				+
17	+		+			+										+										
18																			+							
19																		+								
20		+																			+		+	+		
21																				+						
22			+	+									+	+		+										
23	+	+				+	+		+	+	+	+								+						
24	+	+				+	+		+	+	+	+								+						
25	+	+																								
26			+	+									+	+		+										

Conflicto (X)

Acoplamiento (+)

Redundancia (O)

Independencia

En la matriz de la tabla 4.60 no pueden existir conflictos ni redundancias. Un conflicto supondría que un requisito de software no se podría implementar al existir el requisito de software con el que se encuentra en conflicto, por lo que antes de la implementación se debe solucionar este conflicto. La redundancia supone que dos requisitos de software son iguales por lo que uno de ellos ha de ser eliminado.

4.2.3 Análisis de bases de datos

En esta sección se analizarán las bases de datos de necesarias para el almacenamiento de información requerida por el sistema.

Para almacenar y manejar toda la información del sistema será necesario el uso de bases de datos. Se requerirá en total dos bases de datos, una para cada subsistema: una para el servidor y otra para el terminal móvil.

4.2.3.1 Aplicación móvil

En la base de datos del dispositivo se almacenarán los datos del registro del propietario, la configuración de la aplicación, la lista de sensores que posea el dispositivo, y los datos que se monitoricen cuando el dispositivo no posea conexión a Internet.

En la ilustración 4.3 se puede apreciar el modelo entidad-relación de la base de datos de la aplicación móvil.

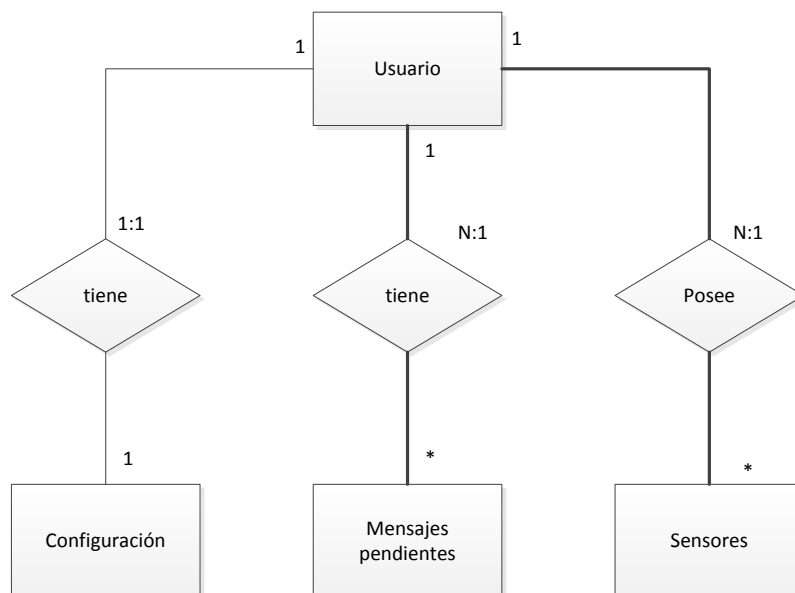


ILUSTRACIÓN 4.3: MODELO ENTIDAD-RELACIÓN DE LA BASE DE DATOS DE LA APLICACIÓN MÓVIL

Como se observa serán necesarias 4 entidades: Usuario, Configuración, Sensores y Mensajes Pendientes. A continuación se describirán estas entidades.

TABLA 4.62: ENTIDAD USUARIO MÓVIL

Nombre	Usuario
Objeto	Almacena los datos de acceso del usuario en la aplicación e información sobre el dispositivo.
Alcance	La persona propietaria del dispositivo.
Clave primaria	La cadena de texto que almacena el correo del usuario.

TABLA 4.63: ENTIDAD CONFIGURACIÓN

Nombre	Configuración
Objeto	Almacena la configuración del usuario para la aplicación.
Alcance	El estado de la monitorización introducido por el usuario.
Clave primaria	Id único.

TABLA 4.64: ENTIDAD SENSORES

Nombre	Sensores
Objeto	Almacena la información sobre los sensores del dispositivo.
Alcance	Los diferentes elementos del dispositivo que toman valores del entorno.
Clave primaria	Id único.

TABLA 4.65: ENTIDAD MENSAJES PENDIENTES

Nombre	Mensajes pendientes
Objeto	Almacena los mensajes pendientes de enviar al servidor.
Alcance	Los datos de la monitorización en un determinado momento en el que el dispositivo no posee conexión a internet.
Clave primaria	Id único.

4.2.3.2 Servidor web

En el servidor web será necesario almacenar la información de todos los usuarios que se registren en la aplicación móvil. La información que se almacenará será los datos de

acceso del usuario, los dispositivos monitorizados, y los datos de los diferentes sensores de cada dispositivo.

En la ilustración 4.4 se puede apreciar el modelo entidad-relación de la base de datos del servidor Web.

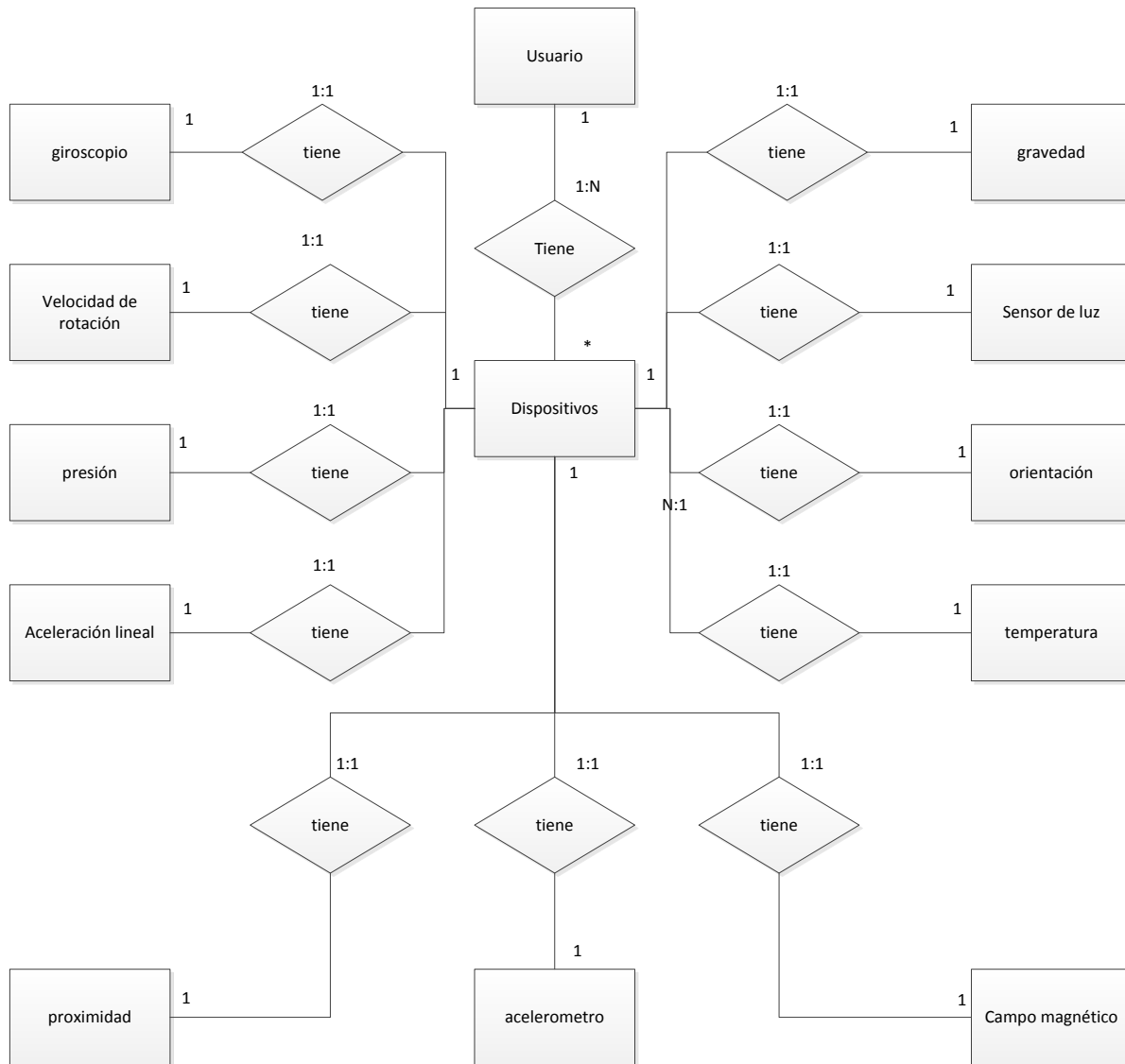


ILUSTRACIÓN 4.4: MODELO ENTIDAD-RELACIÓN DE LA BASE DE DATOS DEL SERVIDOR

A continuación se procederá a definir las propiedades de las diferentes entidades de la base de datos.

TABLA 4.66: ENTIDAD USUARIO SERVIDOR

Nombre	Usuario
Objeto	Almacena los datos de acceso del usuario a la aplicación
Alcance	La persona propietaria del dispositivo monitorizado.
Clave primaria	La cadena que almacena el correo del usuario.

TABLA 4.67: ENTIDAD DISPOSITIVOS

Nombre	Dispositivos
Objeto	Almacena la información sobre el dispositivo monitorizado.
Alcance	El elemento que envía datos al servicio web.
Clave primaria	Las cadenas que almacenan el correo del propietario y el modelo del dispositivo.

TABLA 4.68: ENTIDAD GIROSCOPIO

Nombre	Giroscopio
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide la rotación del globo terrestre
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.69: ENTIDAD VELOCIDAD DE ROTACIÓN

Nombre	Velocidad de rotación
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide la velocidad de rotación del dispositivo.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.70: ENTIDAD PRESIÓN

Nombre	Presión
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide la presión a la que está sometido el terminal móvil.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.71: ENTIDAD ACCELERACIÓN LINEAL

Nombre	Aceleración lineal
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide la aceleración lineal a la que se somete el terminal móvil.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.72: ENTIDAD PROXIMIDAD

Nombre	Proximidad
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide la cercanía a la que se encuentra un elemento del terminal móvil.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.73: ENTIDAD ACCELERÓMETRO

Nombre	Acelerómetro
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide la aceleración del terminal móvil.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.74: ENTIDAD CAMPO MAGNÉTICO

Nombre	Campo magnético
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide el campo magnético a la que se somete el terminal móvil.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.75: ENTIDAD TEMPERATURA

Nombre	Temperatura
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide la temperatura.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.76: ENTIDAD ORIENTACIÓN

Nombre	Orientación
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que marca la orientación del terminal móvil.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.77: ENTIDAD SENSOR DE LUZ

Nombre	Sensor de luz
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide el nivel de luz que recibe el terminal móvil.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

TABLA 4.78: ENTIDAD GRAVEDAD

Nombre	Gravedad
Objeto	Almacena los datos monitorizados del sensor.
Alcance	Dispositivo que mide la gravedad a la que se somete el terminal móvil.
Clave primaria	El conjunto del identificador del dispositivo monitorizado y la fecha en la que se han tomado los datos.

5 Diseño

En este capítulo se explicará el diseño del sistema que se pretende desarrollar en este proyecto. El objetivo del diseño es definir detalladamente la arquitectura y los componentes del sistema.

Para comenzar se comentará el lenguaje de programación que se utilizará para desarrollar el sistema. En segundo lugar se procederá a diseñar la estructura de las bases de datos. Se seguirá con el diseño de la comunicación entre los dos subsistemas. Se concluirá con el diseño de las interfaces de usuario de los subsistemas.

5.1 Lenguaje de programación

El sistema que se pretende desarrollar se puede dividir en dos subsistemas, un servidor web y una aplicación móvil. Cada uno de estos subsistemas se deberá desarrollar con un lenguaje de programación acorde con el entorno en el que deberán funcionar.

La elección de los lenguajes se ha realizado teniendo en cuenta principalmente el conocimiento y la familiaridad del equipo de desarrollo con los lenguajes de programación, lo que supondrá una considerable disminución de tiempo de desarrollo ya que no será necesario tiempo extra para el aprendizaje de nuevos lenguajes.

5.1.1 Servidor web

En el servidor web se almacenarán tanto la página Web con la que interactuarán los usuarios, como los archivos del Web Service que se encargarán de recibir y manejar los datos que envíen las aplicaciones de los dispositivos móviles.

Para desarrollar la página web se utilizarán los lenguajes HTML y CSS para maquetar la información que se mostrará al usuario cuando navegue por la página Web. Para las diferentes funcionalidades de la página, como consultas a la base de datos o el control de sesión, se utilizarán los lenguajes de programación PHP y JavaScript.

El Web Service encargado de recibir y manejar los datos que envíen los dispositivos móviles se programará en su mayoría en PHP.

HTML es un lenguaje de marcado que predomina en la elaboración de páginas webs. Se utiliza para describir y traducir la estructura y la información en forma de texto, así como para completar el texto con objetos tales como imágenes.

CSS es un lenguaje utilizado para definir la presentación de un documento estructurado escrito en HTML o XML. EL W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores.

JavaScript es un lenguaje de programación interpretado. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas Web dinámicas.

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se utiliza principalmente para la interpretación del lado del servidor.

5.1.2 Aplicación móvil

La aplicación móvil se programará en el lenguaje Java dado que es el lenguaje utilizado para el desarrollo de aplicaciones en la plataforma Android. Google proporciona a los desarrolladores gratuitamente el Android Software Development Kit, también conocido como Android SDK.

Java es un lenguaje de programación de alto nivel orientado a objetos. En Android no existe una máquina virtual Java, el bytecode no es ejecutado, sino que en primer lugar se compila en un ejecutable Dalvik, una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con baterías y tienen memoria y procesadores limitados.

5.2 Base de datos

En este apartado se diseñarán las estructuras de las bases de datos que se necesitan para almacenar la información que requiere el sistema.

Como ya se ha mencionado anteriormente en la sección de análisis se utilizarán dos bases de datos, una en la aplicación móvil y otra en el servidor web. A continuación se diseñarán cada una por separado.

5.2.1 Aplicación móvil

Para el almacenamiento y gestión de datos en los dispositivos móviles se utilizará SQLite, un motor de bases de datos que incorpora el sistema operativo Android y los principales sistemas operativos móviles por su reducido tamaño. Al ser Open Source no supondrá un encarecimiento del presupuesto. Entre sus características se incluyen:

- No necesita configuración, ni tras la instalación ni para el mantenimiento.
- No requiere de un servidor. Se utiliza embebida dentro de otra aplicación.
- Utiliza un solo fichero para almacenar los datos.
- Los ficheros de datos son multiplataforma.
- Es muy compacta.
- Utiliza Manifest Typing, permite almacenar cualquier valor de cualquier tipo.
- Utiliza registros de longitud variable, cada dato almacenado en la base de datos ocupará su tamaño real y no el reservado.

En la ilustración 5.1 se puede apreciar la estructura diseñada para la base de datos de la aplicación móvil.

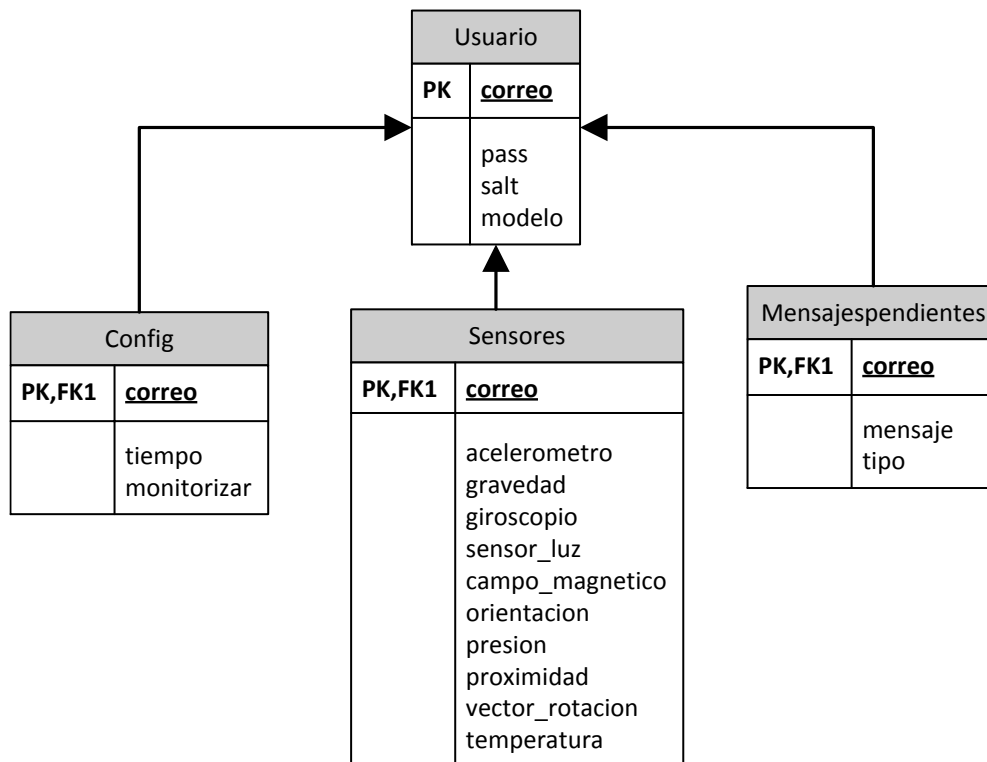


ILUSTRACIÓN 5.1: DIAGRAMA DE LA BASE DE DATOS DE LA APLICACIÓN MÓVIL

A continuación se explicarán los diferentes atributos de las tablas de la base de datos de la aplicación móvil.

TABLA 5.1: COLUMNAS DE LA TABLA USUARIO

Usuario	
correo	En esta columna se guarda el correo electrónico del usuario.
pass	En esta columna se guarda la función resumen de la contraseña del usuario.
salt	En esta columna se guarda la cadena aleatoria utilizada para la creación de la función resumen de la contraseña del usuario.
modelo	En esta columna se guarda el modelo del dispositivo móvil utilizado para la identificación.

TABLA 5.2: COLUMNAS DE LA TABLA CONFIG

Config	
Correo	En esta columna se guarda el correo electrónico del usuario.
Tiempo	Periodo de tiempo de la monitorización
monitorizar	Estado de la monitorización del dispositivo. Marca si esta está activa o desactivada.

TABLA 5.3: COLUMNAS DE LA TABLA SENSORES

Sensores	
Correo	En esta columna se guarda el correo electrónico del usuario.
acelerómetro	Columna que indica si el dispositivo posee acelerómetro o no.
Gravedad	Columna que indica si el dispositivo posee sensor de gravedad o no.
Giroscopio	Columna que indica si el dispositivo posee giroscopio o no.
Sensor_luz	Columna que indica si el dispositivo posee sensor de luz o no.
Aceleración_lineal	Columna que indica si el dispositivo posee sensor de aceleración lineal o no.
Campo_magnetico	Columna que indica si el dispositivo posee sensor de campo magnético o no.
Orientación	Columna que indica si el dispositivo posee sensor de orientación o no.
Presión	Columna que indica si el dispositivo posee sensor de presión o no.
Proximidad	Columna que indica si el dispositivo posee sensor de proximidad o no.
Vector_rotacion	Columna que indica si el dispositivo posee sensor de vector de rotación o no.
temperatura	Columna que indica si el dispositivo posee sensor de temperatura o no.

TABLA 5.4: COLUMNAS DE LA TABLA MENSAJESPENDIENTES

Mensajespendientes	
Correo	En esta columna se guarda el correo electrónico del usuario.
mensajes	Columna donde se guardan los mensajes de la monitorización pendientes de enviar al servidor.
Tipo	Tipo de mensaje pendiente, indica si el mensaje es de monitorización o de cambio de periodo.

5.2.2 Servidor web

Para el almacenamiento de los datos en el servidor web se utilizará MySQL por su sencillez facilitará el insertado y extracción de datos de la base de datos. Al ser Open Source no supondrá un encarecimiento del presupuesto. Entre las principales características de MySQL se encuentran:

- Aprovechamiento de la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de APIs en gran cantidad de lenguajes: C, C++, Java, PHP.
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Se puede descargar su código fuente.
- Gran rapidez y facilidad de uso.
- Fácil instalación y configuración.

En la ilustración 5.2 se puede apreciar el diagrama diseñado para la base de datos del servidor.

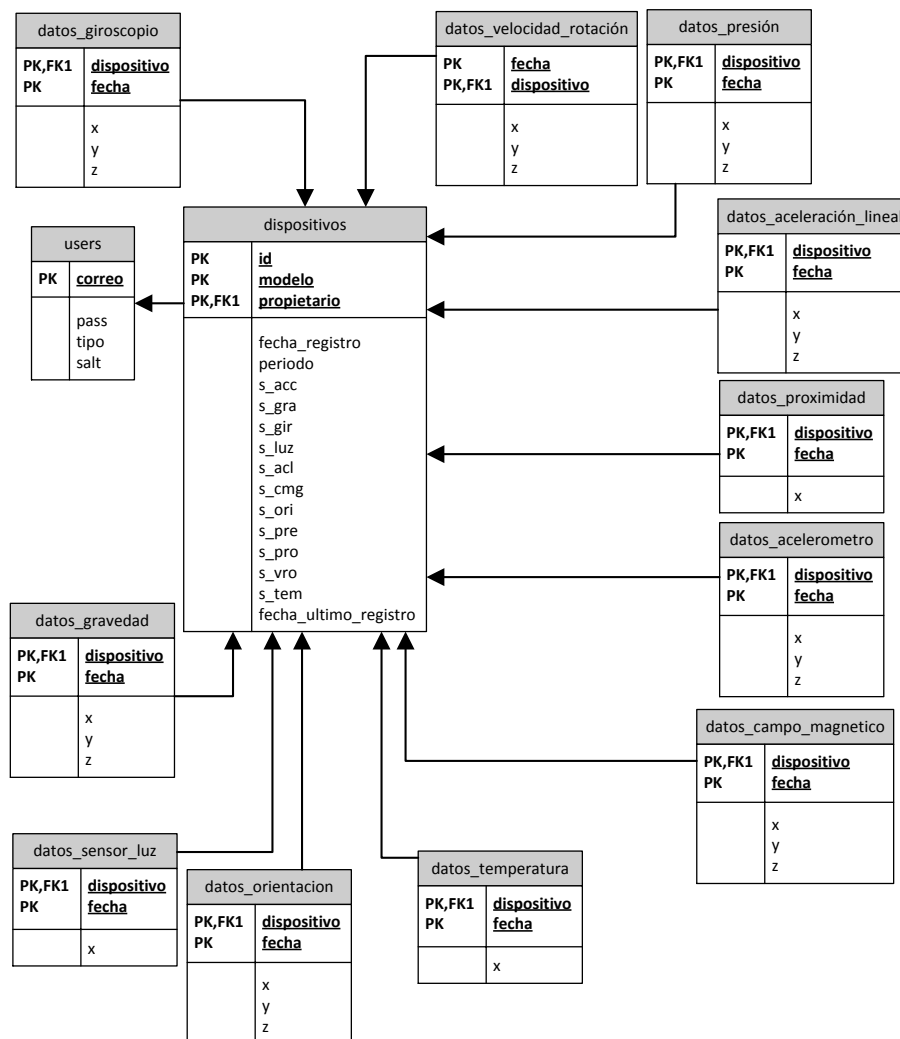


ILUSTRACIÓN 5.2: DIAGRAMA DE LA BASE DE DATOS DEL SERVIDOR

A continuación se definirán las columnas de las tablas de la base de datos del servidor.

TABLA 5.5: COLUMNAS DE LA TABLA USERS

users	
correo	En esta columna se guarda el correo electrónico del usuario.
pass	En esta columna se guarda la función resumen de la contraseña del usuario.
tipo	En esta columna se guarda el indicador de tipo de usuario. Marca si el usuario es un usuario normal o un usuario administrador.
salt	En esta columna se guarda la cadena aleatoria utilizada para la creación de la función resumen de la contraseña del usuario.

TABLA 5.6: COLUMNAS DE LA TABLA DISPOSITIVOS

dispositivos	
id	Identificador del dispositivo.
modelo	Modelo del dispositivo.
propietario	Usuario propietario del dispositivo.
fecha_registro	Fecha en la que se registro el dispositivo.
periodo	Periodo de la monitorización
s_acc	Columna que indica si el dispositivo posee acelerómetro o no.
s_gra	Columna que indica si el dispositivo posee sensor de gravedad o no.
s_gir	Columna que indica si el dispositivo posee giroscopio o no.
s_luz	Columna que indica si el dispositivo posee sensor de luz o no.
s_acl	Columna que indica si el dispositivo posee sensor de aceleración lineal o no.
s_cmg	Columna que indica si el dispositivo posee sensor de campo magnético o no.
s_ori	Columna que indica si el dispositivo posee sensor de orientación o no.
s_pre	Columna que indica si el dispositivo posee sensor de presión o no.
s_pro	Columna que indica si el dispositivo posee sensor de proximidad o no.
s_vro	Columna que indica si el dispositivo posee sensor de vector de rotación o no.

s_tem	Columna que indica si el dispositivo posee sensor de temperatura o no.
Fecha_ultimo_registro	Indica la fecha de la última monitorización registrada.

TABLA 5.7: COLUMNAS DE LA TABLA DATOS_GIROSCOPIO

datos_giroscopio	
Dispositivo	Dispositivo monitorizado
Fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x
y	Valor del sensor en el eje y
z	Valor del sensor en el eje z

TABLA 5.8: COLUMNAS DE LA TABLA DATOS_VELOCIDAD_ROTACION

datos_velocidad_rotacion	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x
y	Valor del sensor en el eje y
z	Valor del sensor en el eje z

TABLA 5.9: COLUMNAS DE LA TABLA DATOS_PRESION

datos_presion	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x
y	Valor del sensor en el eje y
z	Valor del sensor en el eje z

TABLA 5.10: COLUMNAS DE LA TABLA DATOS_ACELERACION_LINEAL

datos_aceleracion_lineal	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x
y	Valor del sensor en el eje y
z	Valor del sensor en el eje z

TABLA 5.11: COLUMNAS DE LA TABLA DATOS_PROXIMIDAD

datos_proximidad	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x

TABLA 5.12: COLUMNAS DE LA TABLA DATOS_ACELEROMETRO

datos_acelerometro	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x
y	Valor del sensor en el eje y
z	Valor del sensor en el eje z

TABLA 5.13: COLUMNAS DE LA TABLA DATOS_CAMPO_MAGNETICO

datos_campo_magnetico	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x
y	Valor del sensor en el eje y
z	Valor del sensor en el eje z

TABLA 5.14: COLUMNAS DE LA TABLA DATOS_TEMPERATURA

datos_temperatura	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x

TABLA 5.15: COLUMNAS DE LA TABLA DATOS_ORIENTACION

datos_orientacion	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x
y	Valor del sensor en el eje y
z	Valor del sensor en el eje z

TABLA 5.16: COLUMNAS DE LA TABLA DATOS_SENSOR_LUZ

datos_sensor_luz	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x

TABLA 5.17: COLUMNAS DE LA TABLA DATOS_GRAVEDAD

datos_gravedad	
dispositivo	Dispositivo monitorizado
fecha	Fecha de los datos monitorizados
x	Valor del sensor en el eje x
y	Valor del sensor en el eje y
z	Valor del sensor en el eje z

5.3 Comunicaciones

EL proyecto que se está desarrollando se trata de un sistema distribuido, lo que significa que se compone, como mínimo, de dos subsistemas que deberán comunicarse de cierta manera. En este apartado se diseñará el sistema de comunicación de estos subsistemas.

Como se ha mencionado anteriormente en el documento, los subsistemas en los que se divide el sistema a desarrollar son: un servidor Web, con una página Web y un servicio, y una aplicación móvil, con una interfaz y un servicio. La comunicación principalmente se dará entre el servicio web y el servicio de la aplicación móvil.

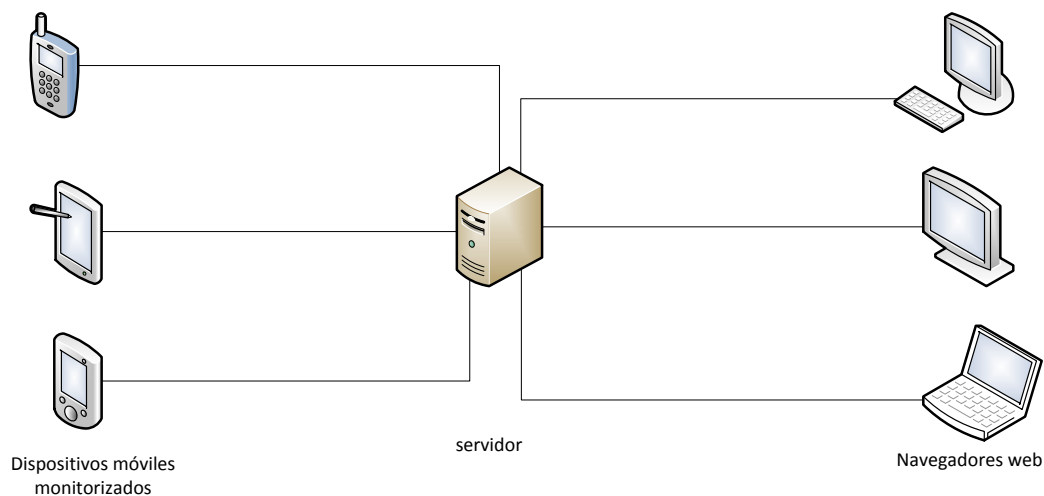


ILUSTRACIÓN 5.3: COMUNICACIÓN DEL SISTEMA

En la ilustración 5.3 se puede apreciar un esquema de las comunicaciones que se producen en el sistema. Los dispositivos móviles se comunicarán con el servidor web enviando los datos del dispositivo. El servidor gestionará y almacenará estos datos, los cuales podrán ser consultados a través de los navegadores Web por los usuarios que visiten la Web almacenada en el servidor. La comunicación que concierne a este diseño es la dada entre los dispositivos móviles y el servidor.

La comunicación se dará principalmente de la aplicación móvil al servidor. El servidor no enviará mensajes a la aplicación salvo por los mensajes de respuesta a los mensajes recibidos.

Los mensajes que la aplicación móvil enviará al servidor serán:

- Mensaje de registro
- Mensaje de monitorización
- Mensaje de cambio de periodo
- Mensaje de cambio de contraseña

En la ilustración 5.4 se puede apreciar el diagrama de secuencia común para las comunicaciones de registro, cambio de periodo y cambio de contraseña. En la función de monitorización no participa el usuario por lo que su diagrama se verá más adelante.

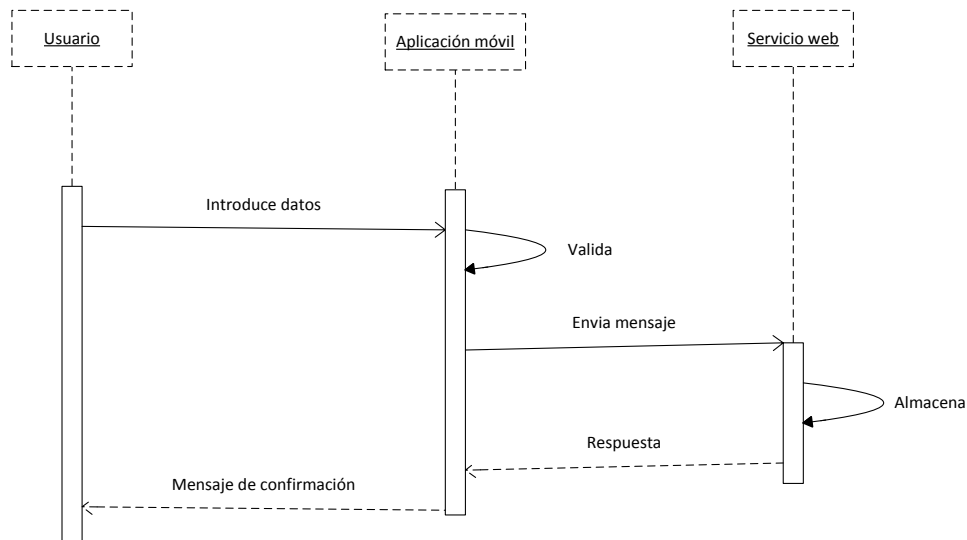


ILUSTRACIÓN 5.4: DIAGRAMA DE SECUENCIA DE LAS COMUNICACIONES

Tal y como se ve en la ilustración anterior, el usuario es siempre quien inicia la comunicación entre la aplicación móvil y el servidor introduciendo datos a través de la interfaz de la aplicación móvil. La aplicación se encargará de validar los datos introducidos, en caso de que sean correctos enviará estos datos al servidor para que los almacene y esperará la respuesta de este. Una vez recibida la respuesta del servidor informará al usuario del resultado de la operación mediante un mensaje visual.

A continuación se procederá a explicar con más detalle estas funciones por separadas.

5.3.1 Registro

Este es el primer mensaje que se intercambiarán la aplicación móvil y el servidor web. Se producirá en cuanto el usuario introduzca los datos solicitados en el formulario de registro. La aplicación móvil se encargará de validar estos datos, en caso de que alguno no sea válido informará al usuario y le pedirá que vuelva a introducirlo. En caso de que todos los valores introducidos sean correctos la aplicación procederá a tomar el resto de valores necesarios para enviar al servidor. Los datos que enviará la aplicación al servidor para que realice el registro serán los siguientes.

- Correo: El correo que ha introducido el usuario en el registro.

- Pass: La función resumen de la contraseña introducida por el usuario sumado al Salt aleatorio.
- Periodo: El valor del periodo introducido por el usuario.
- Modelo: El modelo del dispositivo.
- Fecha: La fecha del momento en el que se va a enviar el mensaje.
- Salt: Cadena aleatoria calculada para la función resumen de la contraseña.
- Acc: Un valor numérico que indica si el dispositivo posee acelerómetro no. Un 1 si sí lo tiene o un 0 si no lo tiene.
- Gra: Un valor numérico que indica si el dispositivo posee sensor de gravedad o no. Un 1 si sí lo tiene o un 0 si no lo tiene.
- Gir: Un valor numérico que indica si el dispositivo posee giroscopio o no. Un 1 si sí lo tiene o un 0 si no lo tiene.
- Luz: Un valor numérico que indica si el dispositivo posee sensor de luz o no. Un 0 si no tiene sensor o un número mayor que cero si tiene, este número marca el valor máximo que puede tomar el sensor.
- Acl: Un valor numérico que indica si el dispositivo posee sensor de aceleración o no. Un 1 si sí lo tiene o un 0 si no lo tiene.
- Cmg: Un valor numérico que indica si el dispositivo posee sensor de campo magnético o no. Un 1 si sí lo tiene o un 0 si no lo tiene.
- Ori: Un valor numérico que indica si el dispositivo posee sensor de orientación o no. Un 1 si sí lo tiene o un 0 si no lo tiene.
- Pre: Un valor numérico que indica si el dispositivo posee sensor de presión o no. Un 0 si no tiene sensor o un número mayor que cero si tiene, este número marca el valor máximo que puede tomar el sensor.
- Pro: Un valor numérico que indica si el dispositivo posee sensor de proximidad o no. Un 0 si no tiene sensor o un número mayor que cero si tiene, este número marca el valor máximo que puede tomar el sensor.
- Vro: Un valor numérico que indica si el dispositivo posee sensor de vector de rotación o no. Un 1 si sí lo tiene o un 0 si no lo tiene.
- Tem: Un valor numérico que indica si el dispositivo posee sensor de temperatura o no. Un 1 si sí lo tiene o un 0 si no lo tiene.

El servidor almacenará estos datos en la base de datos y responderá con "OK" si ha tratado correctamente los datos recibidos. Si se ha producido algún error responderá con un "KO".

5.3.2 Cambio de periodo

Este mensaje se enviará cuando el usuario cambie el periodo a través de las opciones de la aplicación móvil. Los datos que enviará la aplicación al servidor serán:

- Correo: Correo del usuario.
- Modelo: Modelo del dispositivo que envía el mensaje.
- tiempo: Nuevo valor para el periodo.

El servidor responderá con “OK” si ha tratado correctamente los datos recibidos. Si se ha producido algún error responderá con un “KO”.

5.3.3 Cambio de contraseña

Este mensaje se enviará cuando el usuario cambie la contraseña a través de las opciones de la aplicación móvil. Los datos que enviará la aplicación al servidor serán:

- correo: Correo del usuario.
- pass: La función resumen de la antigua contraseña.
- npass: La función resumen de la nueva contraseña.
- salt: La nueva cadena aleatoria calculada para la función resumen de la nueva contraseña.

El servidor responderá con “OK” si ha tratado correctamente los datos recibidos. Si se ha producido algún error responderá con un “KO”.

5.3.4 Monitorización

La monitorización, tal y como se ha mencionado con anterioridad, se realiza sin la intervención del usuario. Este sólo influye en la activación o desactivación del envío de estos mensajes. Pero si la monitorización está activa estos se envían sin más intervención del usuario. Esta función se realiza en el servicio de la aplicación móvil.

En la ilustración 5.5 se puede apreciar el diagrama de secuencia del servicio de la aplicación móvil.

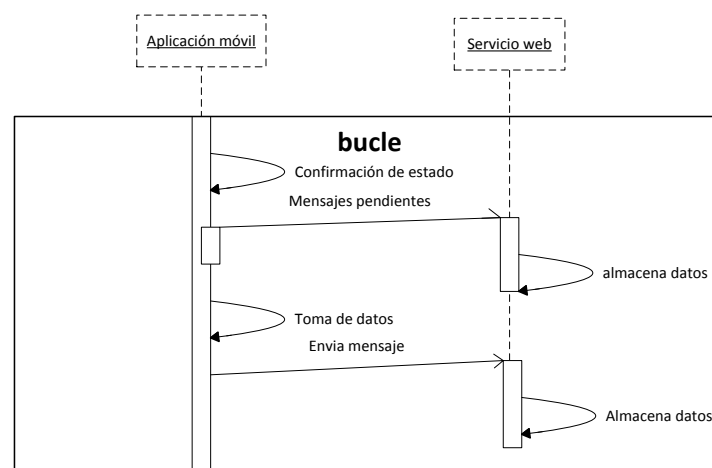


ILUSTRACIÓN 5.5: DIAGRAMA DE SECUENCIA DEL SERVICIO DE LA APLICACIÓN MÓVIL

El servicio puede enviar dos tipos de mensajes de monitorización. El primero con mensajes pendientes y el segundo del actual estado de la monitorización. El bucle se ejecutará cada equis segundos que marcará el usuario en el periodo. Los datos que enviará el servicio al servidor web serán.

- Correo: Correo del usuario.
- modelo: El modelo del dispositivo.
- fecha: fecha de la monitorización.
- acc_x: Valor del acelerómetro en el eje x.
- acc_y: Valor del acelerómetro en el eje y.
- acc_z: Valor del acelerómetro en el eje z.
- gra_x: Valor de en el eje x.
- gra_y: Valor de en el eje y.
- gra_z: Valor de en el eje z.
- gir_x: Valor de en el eje x.
- gir_y: Valor de en el eje y.
- gir_z: Valor de en el eje z.
- luz_x: Valor del sensor de luz
- acl_x: Valor de en el eje x.
- acl_y: Valor de en el eje y.
- acl_z: Valor de en el eje z.
- cmg_x: Valor de en el eje x.
- cmg_y: Valor de en el eje y.
- cmg_z: Valor de en el eje z.
- ori_x: Valor de en el eje x.
- ori_y: Valor de en el eje y.
- ori_z: Valor de en el eje z.
- pre_x: Valor del sensor de presión.
- pro_x: Valor del sensor de proximidad.
- vro_x: Valor de en el eje x.
- vro_y: Valor de en el eje y.
- vro_z: Valor de en el eje z.
- tem_x: Valor del sensor de temperatura.

Para mejor comodidad del manejo de los datos, los valores de los sensores se enviarán por separado.

5.4 Aplicación Servicio

En el dispositivo móvil la toma de datos de los sensores se debe ejecutar sin que el usuario interfiera ni esté al corriente. Esto se realiza en Android mediante un servicio. Los servicios son funciones que se ejecutan en segundo plano y que no interactúan con el usuario.

En la ilustración 5.6 se puede apreciar el diagrama de flujo del servicio Android.

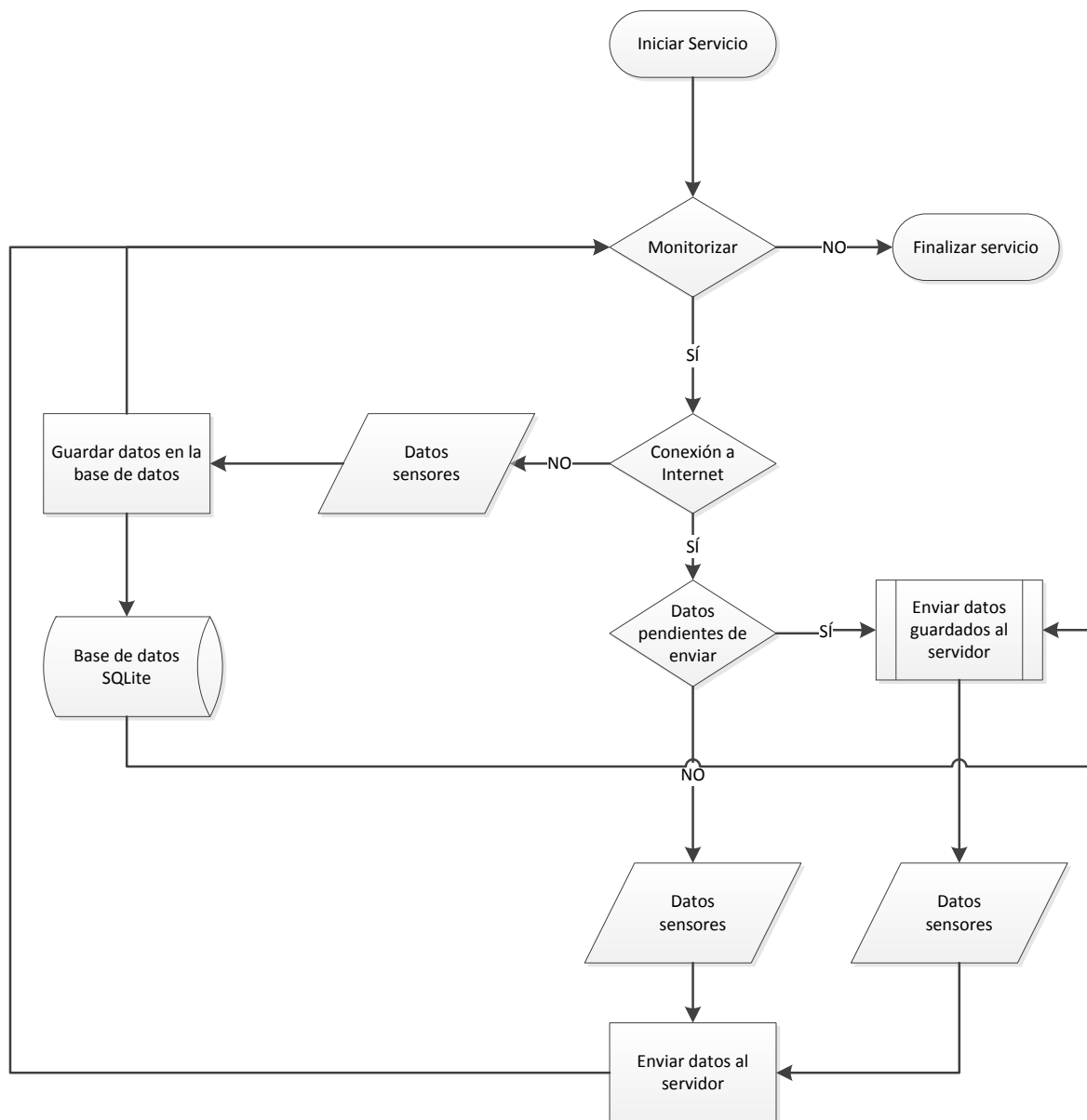


ILUSTRACIÓN 5.6: DIAGRAMA DE FLUJO DEL SERVICIO DE LA APLICACIÓN MÓVIL

Tal y como se especifica en los casos de uso y los requisitos, el usuario podrá decidir si el servicio se ejecuta o no desde la aplicación móvil. Si el usuario cancela la monitorización el servicio lo detectará al ejecutarse y terminará el servicio. Si la monitorización está activada lo primero que comprobará el servicio es la conexión a Internet. En caso de que no se tenga conexión a Internet se tomarán los datos de los sensores y se guardarán en la base de datos. En caso de que sí se tenga conexión a Internet, primero se comprobará si hay mensajes pendientes guardados en la base de datos, si los hay se enviarán en un segundo hilo, si no los haya se tomarán los datos de los sensores y se enviarán servidor Web.

5.5 Interfaces de usuario

En este apartado se definirán las interfaces de usuario de los subsistemas de la aplicación. La interfaz de usuario es el medio con el que el usuario puede comunicarse con la máquina, dispositivo móvil o servidor web dependiendo del subsistema. La interfaz de una aplicación es uno de los puntos más importantes del sistema de cara al público, por lo que se debe hacer una interfaz sencilla y fácil de comprender.

5.5.1 Aplicación móvil

Lo más importantes de una interfaz en una aplicación móvil es el aprovechamiento del espacio reducido que otorgan estos dispositivos y el aprovechamiento de funcionalidad táctiles de las pantallas.

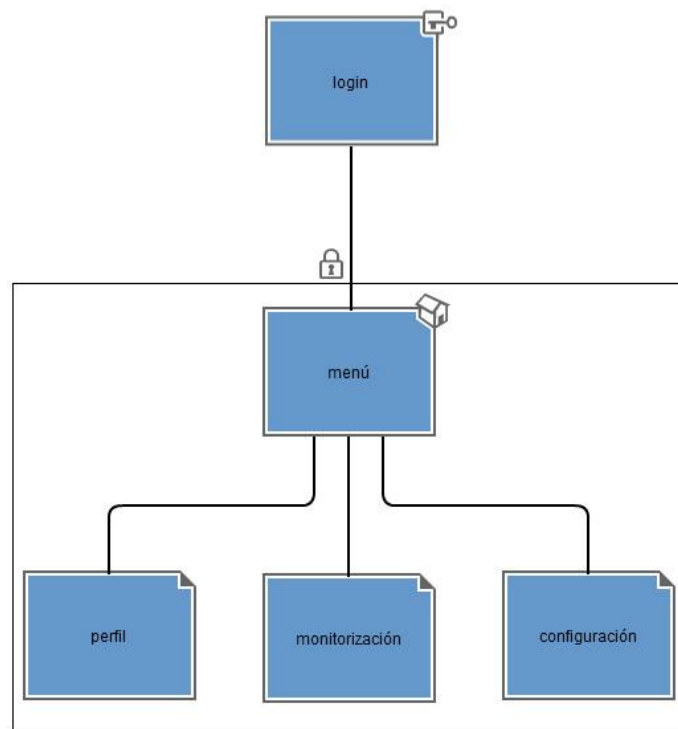


ILUSTRACIÓN 5.7: NAVEGACIÓN ENTRE PANTALLAS DE LA APLICACIÓN MÓVIL

En la ilustración 5.7 se pueden apreciar el número de páginas requeridas y la navegación entre ellas para mostrar toda la información necesaria en la aplicación móvil. Tal y como se aprecia para acceder a las diferentes páginas de la aplicación será necesario pasar a través de una página de identificación.

5.5.1.1 Interfaz login

Esta página se divide en dos vistas dependiendo si el usuario ya ha realizado el registro en la aplicación o no. Si el usuario no se ha registrado se mostrará la ventana de registro, en cambio si ya ha realizado el registro con anterioridad se mostrará la ventana de identificación.

5.5.1.1.1 Registro

En esta ventana se mostrará un formulario el cual el usuario deberá rellenar para poder registrarse en la aplicación, los datos que el usuario deberá introducir serán:

- Un correo electrónico que le identifique.
- Una contraseña de acceso.
- Un valor numérico que sirva como periodo de la monitorización

En la ilustración 5.8 se puede apreciar el diseño de esta interfaz. Como se puede apreciar al formulario de registro le precederá una ilustración que haga de cabecera de la página y un texto explicativo.

ILUSTRACIÓN 5.8: INTERFAZ DE REGISTRO DE LA APLICACIÓN MÓVIL

5.5.1.1.2 Identificación

Esta interfaz será necesaria para que el usuario se identifique y pueda acceder a la aplicación. Para identificarse el usuario deberá introducir el correo electrónico y la contraseña introducidas al realizar el registro en la interfaz anteriormente descrita.

En la ilustración 5.9 se puede apreciar el diseño de esta interfaz.



The image shows a user login interface for the 'Android monitor' application. At the top, there is a placeholder for a logo, represented by a rectangle with a dashed 'X' and the word 'logo' in the center. Below this, the text 'identifíquese en Android monitor' is displayed. The interface includes two input fields: one for 'correo' (email) and one for 'contraseña' (password). Both fields are currently empty. Below the password field is a button labeled 'aceptar' (accept).

ILUSTRACIÓN 5.9: INTERFAZ DE USUARIO DEL LOGIN DE LA APLICACIÓN MÓVIL

Como se puede apreciar el diseño es idéntico a la interfaz de registro salvo en que se excluye la solicitud del valor del periodo que no es necesario.

5.5.1.2 Interfaz menú

El menú es la interfaz principal de la aplicación móvil, una vez pasado el control de acceso el usuario es redirigido a esta pantalla. El menú es una pantalla sencilla que sólo sirve para elegir que es lo que se quiere hacer en la aplicación. Desde esta pantalla se podrá navegar al resto de pantallas de la aplicación: el perfil, la monitorización y la configuración; Por lo que en esta interfaz deberán figurar los tres enlaces a cada una de estas pantallas.

En la ilustración 5.10 se puede apreciar el diseño de la interfaz de esta página.

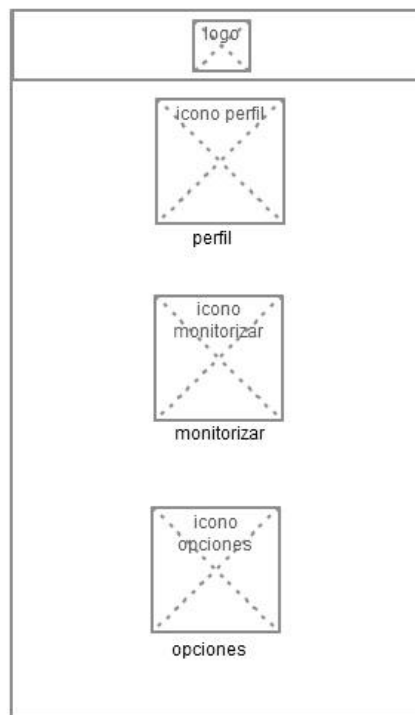


ILUSTRACIÓN 5.10: INTERFAZ DEL MENÚ DE LA APLICACIÓN MÓVIL

Como se puede apreciar en el diseño se busca la simetría situando un icono por enlace en fila uno debajo de otro. En la parte superior se situará una barra que se mantendrá en las siguientes vistas y que servirá como navegación en las páginas interiores.

5.5.1.3 Interfaz perfil

En la página de perfil se deben mostrar los datos de la cuenta del usuario, los datos que se mostrarán en esta pantalla serán: el correo del usuario, el modelo del teléfono, y el periodo de la monitorización.

En la ilustración 5.11 se puede apreciar el diseño de esta pantalla.

valver	logo
correo	
	correo
modelo	
	modelo
periodo (s)	
	periodo s

ILUSTRACIÓN 5.11: INTERFAZ DEL PERFIL DE LA APLICACIÓN MÓVIL

Tal y como se observa en la ilustración anterior los tres datos que se deberán mostrar en esta página se mostrarán en fila uno debajo de otro. En primer lugar se mostrará una barra con el nombre del dato que se mostrará a continuación, seguidamente y alineado a la derecha se mostrará el valor de este dato. En la parte izquierda de la barra de navegación se incluye un botón que servirá para que la aplicación vuelva a mostrar la pantalla del menú.

5.5.1.4 Interfaz monitorización

En esta pantalla se mostrarán la lista de sensores y los valores que se han monitorizado de cada uno de estos sensores.

En la ilustración 5.12 se puede apreciar el diseño de esta pantalla.

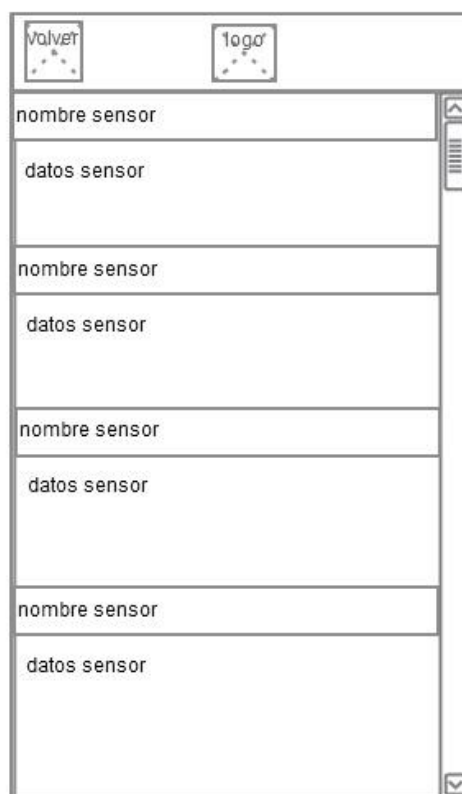


ILUSTRACIÓN 5.12: INTERFAZ DE LA MONITORIZACIÓN DE LA APLICACIÓN MÓVIL

Como se puede observar el diseño compartirá el estilo que la pantalla de perfil, mostrando en un primer lugar en una franja el nombre del sensor y a continuación los valores de los sensores. Al ser bastantes datos será necesario que la pantalla cuente con un desplazamiento vertical para poder mostrar todos los datos. La barra de la parte superior vuelve a contar con el botón de volver al menú en la parte izquierda.

5.5.1.5 Interfaz configuración

En la interfaz de configuración u opciones se mostrarán los datos de la aplicación que el usuario será capaz de modificar. Estos datos son el estado de la monitorización, el periodo de la monitorización y la contraseña de acceso a la aplicación.

En la ilustración 5.13 se puede apreciar el diseño de esta pantalla.

The image shows a wireframe of a mobile application configuration screen. At the top, there is a header bar containing two logos: 'Valver' on the left and 'logor' on the right. Below the header, the screen is divided into three main sections. The first section is titled 'Monitorizar' and contains a single checkbox. The second section is titled 'Cambiar periodo' and contains a text input field labeled 'periodo actual' and a button labeled 'enviar'. The third section is titled 'Cambiar contraseña' and contains two text input fields, one labeled 'actual contraseña' and the other 'nueva contraseña', followed by a button labeled 'enviar'.

ILUSTRACIÓN 5.13: INTERFAZ DE LA CONFIGURACIÓN DE LA APLICACIÓN MÓVIL

Tal y como se observa se sigue mantenido el estilo de diseño de la páginas anteriormente descritas. El estado de la monitorización sólo puede tomar dos valores: activada o desactivada; Por esto el botón para cambiar el estado consistirá en un botón que mostrará el estado actual de la monitorización que cambiará al pulsarlo.

Para cambiar el periodo será necesario un cuadro de texto donde se mostrará el valor actual del periodo y un botón. Para cambiar el periodo el usuario deberá introducir un nuevo valor en el cuadro de texto y pulsar el botón enviar.

Para cambiar la contraseña serán necesarios dos cuadros de texto y un botón. En el primer cuadro de texto el usuario deberá introducir la contraseña actual, en el segundo la nueva contraseña que desee y a continuación pulsar el botón enviar.

5.5.2 Servidor web

En la ilustración 5.14 se puede observar las diferentes páginas necesarias para la página web y la navegación entre ellas.

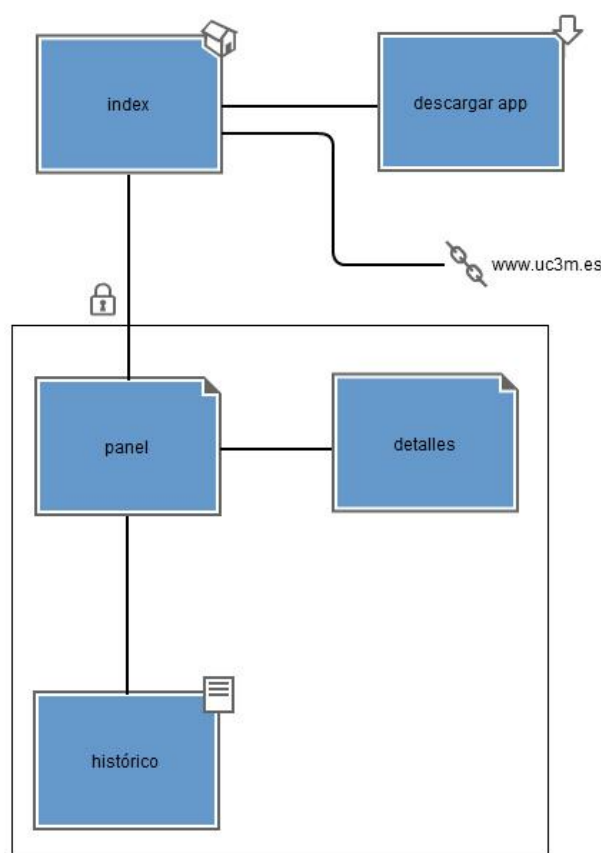


ILUSTRACIÓN 5.14: NAVEGACIÓN ENTRE PANTALLAS DE LA PÁGINA WEB

Tal y como se observa la página principal será la página de índice, desde ella se podrá acceder al enlace que redirigirá a una página externa o a la página de descarga de la aplicación móvil. Desde esta página también se podrá acceder a las funcionalidades de la aplicación web, las páginas panel, detalles e histórico, previa identificación.

5.5.2.1 Interfaz índice

El contenido principal de esta página será un formulario de identificación que se utilizará para acceder a las páginas privadas de la web. También contará con enlaces que servirán para dirigirse a la página de descarga de la aplicación o a la página externa de la universidad. En esta página también será necesario el selector de idioma para cambiar el idioma de la página.

En la ilustración 5.15 se puede apreciar el diseño de esta interfaz.

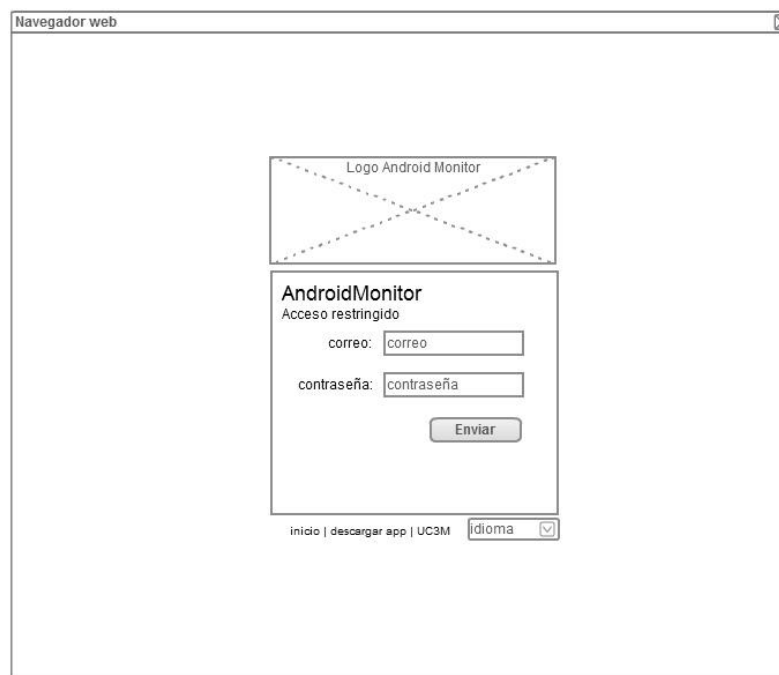


ILUSTRACIÓN 5.15: INTERFAZ DEL ÍNDEX DE LA PÁGINA WEB

Como se puede apreciar, el formulario de identificación se sitúa centrado en la ventana siendo lo primero que es apreciado por el usuario. En la parte inferior del formulario se sitúan los enlaces y el selector de idioma.

5.5.2.2 Interfaz descargar app

El propósito principal de esta página es proporcionar la descarga del archivo de instalación de la aplicación móvil a los usuarios. Esta descarga se podrá realizar tanto por enlace como mediante un código QR.

En la ilustración 5.16 se puede apreciar el diseño de esta interfaz.

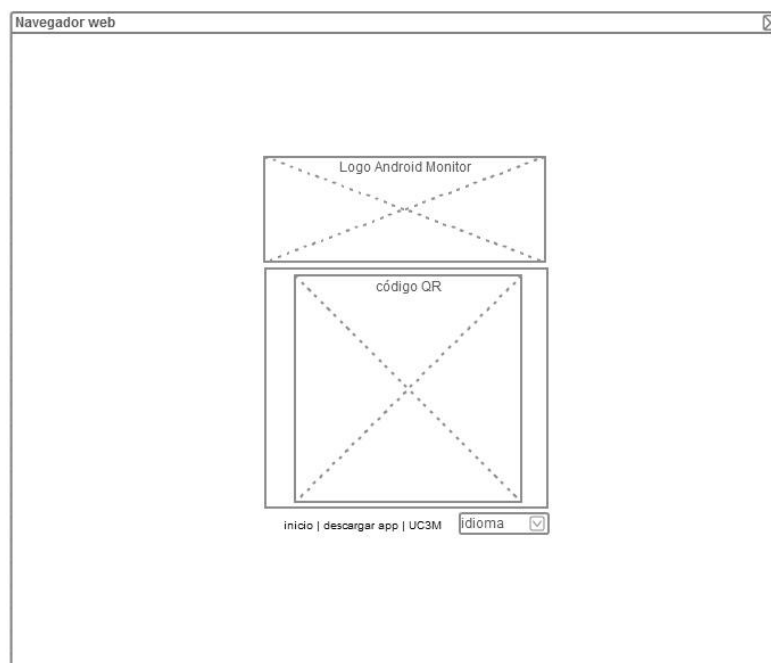


ILUSTRACIÓN 5.16: INTERFAZ DE LA PÁGINA DESCARGAR APP DE LA PÁGINA WEB

Esta página será la única página de la web, junto con la página índice, que será visible sin necesidad de identificación. Por eso se ha decidido que comparta el mismo diseño que la anterior. En el lugar donde en el índice figuraba el formulario de identificación, en esta página se encuentra el código QR y a la vez enlace para descargar la aplicación.

5.5.2.3 Interfaz panel

En esta ventana se mostrarán los datos de los dispositivos monitorizados. La cantidad de información mostrada dependerá del nivel del usuario de sesión. Si es un usuario normal sólo podrá ver los datos de sus dispositivos asociados, en cambio si es un usuario administrador podrá ver todos los dispositivos monitorizados.

En la ilustración 5.17 se puede apreciar el diseño de esta interfaz.

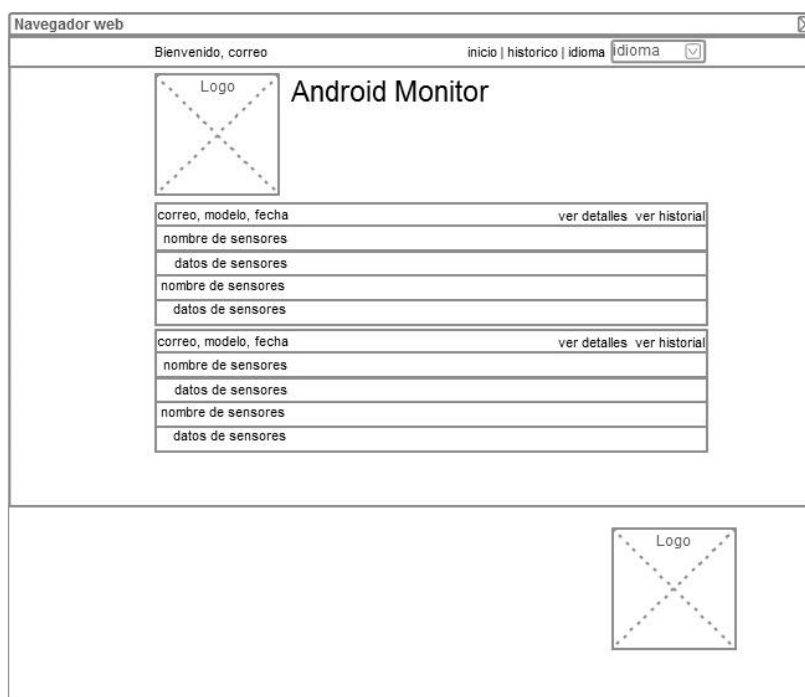


ILUSTRACIÓN 5.17: INTERFAZ DE LA PÁGINA PANEL DE LA PÁGINA WEB

En primer lugar se comentarán los elementos comunes de esta y las siguientes páginas.

En la parte superior se situará una barra. En la parte izquierda de esta se situará un mensaje de bienvenida al usuario. En la parte derecha se situarán los enlaces rápidos de las páginas internas y el selector de idioma con el cual el usuario podrá cambiar el idioma de la web.

Justo debajo de la barra de navegación se situarán el logo y el nombre de la aplicación “Android Monitor” alineados en la parte izquierda.

Al pie de página se situará una sección donde en un futuro se podrán situar más elementos o enlaces. En la parte derecha de esta sección se situará de nuevo el logo de la aplicación.

En la sección central de la página se situará el contenido principal de la página. La información se dividirá en filas y columnas formando una tabla. En la parte izquierda de la primera fila se situarán los datos identificadores del dispositivo monitorizado: el correo del propietario, el modelo del teléfono, y la fecha de la monitorización; En la parte derecha de esta fila se situarán los enlaces para acceder a más información de este dispositivo, ya sea el acceder a los detalles de la monitorización o al historial de esta.

En las siguientes dos filas se situarán uno encima de otro el nombre del sensor y los datos monitorizados de este respectivamente. Al ser bastantes datos no se podrá mostrar toda la información en dos filas por lo que se dividirán por la mitad y la segunda mitad se mostrarán en dos filas más situadas justo debajo.

Esta tabla descrita en el párrafo anterior se repetirá por cada dispositivo monitorizado.

5.5.2.4 Interfaz detalles

En la página de detalles de la monitorización sólo se mostrará la monitorización de un dispositivo.

En la ilustración 5.18 se puede apreciar el diseño de esta interfaz.

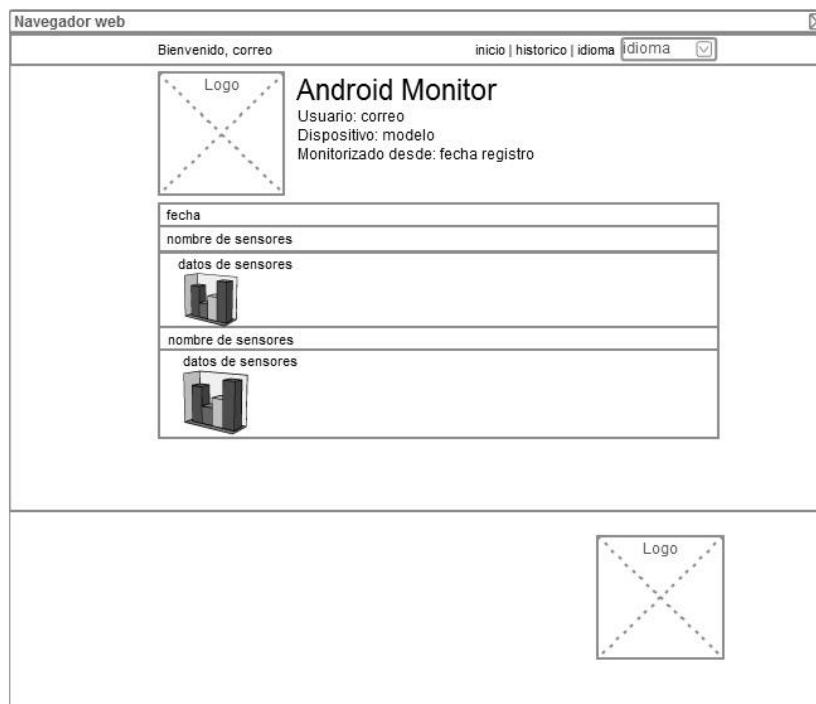


ILUSTRACIÓN 5.18: INTERFAZ DE LA PÁGINA DETALLES DE LA PÁGINA WEB

Los datos de identificación del dispositivo monitorizado se mostrarán a la derecha del logo de la cabecera justo debajo del título. En la primera fila de la tabla de datos donde antes se mostraba la información del dispositivo en esta página sólo se muestra la fecha de la monitorización. En la fila donde muestran los valores de los datos de los sensores ahora aparte se muestra una pequeña ilustración gráfica que ayuda a ilustrar el estado del sensor.

5.5.2.5 Interfaz histórico

El objetivo de esta página es permitir al usuario mediante un formulario consultar los datos de la monitorización de sus dispositivos entre dos fechas. Para esto el formulario deberá contar con:

- Usuario: indicará el propietario del dispositivo que se quiere consultar.

- Dispositivo: El dispositivo que se quiere consultar.
- Fecha desde: Desde que día se quiere consultar.
- Hora desde: Desde que hora se quiere consultar.
- Fecha hasta: Hasta que día se quiere consultar.
- Hora hasta: Hasta que hora se quiere consultar.

En la ilustración 5.19 se puede apreciar el diseño de esta interfaz.

ILUSTRACIÓN 5.19: INTERFAZ DE LA PÁGINA HISTÓRICO DE LA PÁGINA WEB

El formulario de consulta se sitúa a la derecha del logo superior. Se compone de dos selectores, uno para el usuario y otro para el dispositivo, el segundo se encuentra condicionado al primero. Las fechas se introducirán mediante un calendario y las horas mediante un input de hora. Para navegar entre los resultados se contará con un desplazador horizontal que tomará el valor igual al número de resultados obtenidos en la consulta.

La tabla que muestra los datos será similar a la descrita en la Interfaz del panel.

6 Desarrollo

En esta sección se procederá a explicar los detalles de la implementación del proyecto. En primer lugar se explicará los detalles de la implementación del servidor Web y a continuación se explicarán los detalles de la aplicación móvil.

6.1 Servidor web

El servidor web, como ya se ha mencionado anteriormente se divide en dos elementos. El primer elemento que se desarrollará será la parte de servicio Web que se encarga de recibir y manejar los datos que envían los dispositivos. En segundo lugar se describirá la página web que será utilizada por los usuarios para monitorizar distribuidamente sus dispositivos y ver los datos que almacena el servidor de estos.

6.1.1 Servicio Web

En primer lugar se procederá a explicar las funciones que otorga el servidor como servicio web. Estas funciones son las descritas en el punto [5.3 Comunicaciones](#) del capítulo Diseño. La parte que se desarrollará en este punto es la que concierne al servidor, en puntos más adelante se completará desarrollando la parte que debe ejecutar la aplicación móvil.

6.1.1.1 Registro

El registro en el servidor web se realiza en el archivo `registro.php` del directorio `/php_code`. Esta clase se encarga de recibir los datos que envían las aplicaciones móviles y guardarlos en la base de datos. El pseudocódigo necesario para realizar esta operación se describe a continuación.

- Lee y decodifica los datos recibidos y los guarda en variables por separado.
- Si se han leído bien los datos.
 - Establece la conexión con la base de datos.
 - Busca en la base de datos si el usuario existe o no.
 - Si existe el usuario en la base de datos.
 - Si la contraseña enviada es igual a la almacenada
 - Busca en la base de datos si el usuario tiene el dispositivo enviado registrado
 - Si ya tiene un dispositivo igual almacenado
 - Actualiza en la tabla de dispositivos el dispositivo enviado cambiando el periodo y la lista de sensores.
 - Si se ha actualizado correctamente.
 - Devuelve un OK.

- o Si se ha producido algún error al actualizar.
 - Devuelve un KO.
 - Si no tiene un dispositivo igual almacenado.
 - o Inserta en la tabla dispositivos los datos recibidos del dispositivo: el correo, el modelo, la fecha, el periodo, y los datos sobre los sensores.
 - o Si se ha insertado correctamente.
 - Devuelve un OK.
 - o Si se ha producido algún error al insertar.
 - Devuelve un KO.
 - o
 - Si la contraseña enviada es diferente a la almacenada
 - Devuelve un KO.
 - o Si no existe el usuario en la base de datos
 - Inserta en la tabla de usuarios los valores recibidos: el correo, la contraseña y el salt.
 - Si se ha introducido correctamente.
 - Introduce en la tabla de dispositivos los datos del dispositivo recibidos: el correo, el modelo, la fecha, el periodo, y los datos sobre los sensores.
 - Si se ha introducido correctamente.
 - o Devuelve un OK.
 - Si no se ha introducido por algún error.
 - o Devuelve un KO.
 - Si no se ha introducido por algún error.
 - Devuelve un KO.
 - Si no se han leído bien los datos.
 - o Devuelve un KO.

6.1.1.2 Cambio de periodo

El cambio de periodo se realiza en el archivo `cambiarPeriodo.php` que se encuentra en directorio `/php_code`. Este archivo se encarga de cambiar el valor del periodo almacenado para el dispositivo que solicita el cambio. El pseudocódigo necesario para realizar esta operación se describe a continuación.

- Lee y decodifica los datos recibidos y los guarda en variables por separado.
- Si se han leído bien los datos.
 - o Establece conexión con la base de datos.
 - o Busca en la base de datos si el usuario que solicita el cambio de contraseña existe.
 - o Si el usuario existe.
 - Busca en la base de datos si el dispositivo para el que se solicita el cambio existe,
 - Si se encuentra el dispositivo.

- Actualiza en la tabla usuarios el valor del periodo por el nuevo periodo recibido.
 - Si se ha actualizado correctamente.
 - o Devuelve un OK.
 - Si se ha producido algún error al actualizar.
 - o Devuelve un KO.
- Si no se encuentra el dispositivo.
 - Devuelve el salt almacenado.
- o Si el usuario no existe.
 - Devuelve un KO.
- Si no se han leído bien los datos.
 - o Devuelve un KO.

6.1.1.3 Cambio de contraseña

El cambio de contraseña se realiza en el archivo `cambiarPass.php` que se sitúa dentro del directorio `/php_code`. Este archivo se encarga de verificar que la contraseña recibida es idéntica a la almacenada en la base de datos del servidor, en caso de ser así cambia la contraseña almacenada. El pseudocódigo necesario para realizar esta operación se describe a continuación.

- Lee y decodifica los datos recibidos y los guarda en variables por separado.
- Si se han leído bien los datos.
 - o Establece conexión con la base de datos.
 - o Busca en la base de datos si el usuario que solicita el cambio de contraseña existe.
 - o Si el usuario existe.
 - Comprueba que la contraseña almacenada es igual a la contraseña antigua recibida.
 - Si las contraseñas son iguales.
 - Actualiza en la tabla usuarios el valor de la contraseña y el salt por la nueva contraseña y el nuevo salt recibidos donde el correo es igual al recibido.
 - Si se ha actualizado correctamente.
 - o Devuelve un OK.
 - Si se ha producido algún error al actualizar.
 - o Devuelve un KO.
 - Si las contraseñas son distintas.
 - Devuelve KO.
 - o Si el usuario no existe.
 - Devuelve un KO.
- Si no se han leído bien los datos.
 - o Devuelve un KO.

6.1.1.4 Monitorización

El servicio de monitorización se realiza en el archivo `moniSens.php` que se sitúa en el directorio `php_code`. Se encarga de recibir los datos de la monitorización que envía el servicio de la aplicación móvil y almacenarlos en las tablas correspondientes de la base de datos.

- Lee y decodifica los datos recibidos y los guarda en variables por separado.
- Si se han leído bien los datos.
 - Establece conexión con la base de datos.
 - Busca el dispositivo que envía los datos en la tabla dispositivos de la base de datos.
 - Si el dispositivo existe.
 - Se crea una variable de estado inicializándola a 1.
 - Se cambia la variable estado del dispositivo a "online".
 - Si el dispositivo tiene acelerómetro.
 - Se introduce en la tabla de datos del acelerómetro los datos recibidos del acelerómetro, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
 - Si el dispositivo tiene sensor de gravedad.
 - Se introduce en la tabla de datos del sensor de gravedad los datos recibidos del sensor de gravedad, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
 - Si el dispositivo tiene giroscopio.
 - Se introduce en la tabla de datos del giroscopio los datos recibidos del giroscopio, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
 - Si el dispositivo tiene sensor de luz.
 - Se introduce en la tabla de datos del sensor de luz los datos recibidos del sensor de luz, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
 - Si el dispositivo tiene sensor de aceleración lineal.
 - Se introduce en la tabla de datos del sensor de aceleración lineal los datos recibidos del sensor de aceleración lineal, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
 - Si el dispositivo tiene sensor de campo magnético.
 - Se introduce en la tabla de datos del sensor de campo magnético los datos recibidos del

- sensor de campo magnético, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
- Si el dispositivo tiene sensor de orientación.
 - Se introduce en la tabla de datos del sensor de orientación los datos recibidos del sensor de orientación, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
- Si el dispositivo tiene sensor de presión.
 - Se introduce en la tabla de datos del sensor de presión los datos recibidos del sensor de presión, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
- Si el dispositivo tiene sensor de proximidad.
 - Se introduce en la tabla de datos del sensor de proximidad los datos recibidos del sensor de proximidad, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
- Si el dispositivo tiene sensor de vector de rotación.
 - Se introduce en la tabla de datos del sensor de vector de rotación los datos recibidos del sensor de vector de rotación, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
- Si el dispositivo tiene sensor de temperatura.
 - Se introduce en la tabla de datos del sensor de temperatura los datos recibidos del sensor de temperatura, el identificador del dispositivo y la fecha.
 - Si se produce algún error al introducir los datos se cambia una variable de estado a 0.
- Si la variable de estado vale 1.
 - Devuelve OK.
- Si la variable de estado vale 0.
 - Devuelve KO.
- o Si el dispositivo no existe.
 - Devuelve un KO.
- Si no se han leído bien los datos.
 - o Devuelve un KO.

6.1.2 Aplicación Web

En primer lugar se procederá a describir el código común de todas las páginas de la página web, estas funciones son la de cargar y cambiar idioma. A continuación se procederá a describir el funcionamiento de las diferentes páginas que componen la página web. Para cada una de las páginas se describirá el código de sus funcionalidades y la interfaz resultante.

6.1.2.1 Cargar idioma

Lo primero que hará antes de cargar cualquier página será comprobar el lenguaje de la web. Esto se realizará comprobando si existe un valor "iso" pasado como parámetro en la URL de la Web. El algoritmo que realiza esto es el siguiente.

- Si existe el parámetro "iso".
 - o Variable iso igual al parámetro introducido.
- Si no existe el parámetro "iso".
 - o Variable iso igual a "es".
- Si iso es igual a "es" o "en".
 - o Si iso es igual a "es".
 - Se carga el archivo de idioma español del directorio lang/.
 - o Si iso es igual a "en".
 - Se carga el archivo de idioma inglés del directorio lang/.
- Si no.
 - o Se carga el archivo de idioma español del directorio lang/.

El objetivo es cargar las cadenas de texto estáticas de la página web en el idioma correcto almacenadas como variables en unos documentos externos.

6.1.2.2 Cambiar idioma

El cambio de idioma se realizará mediante un selector situado en todas las páginas. El selector cuenta con dos opciones, español e inglés, una de ellas estará siempre seleccionada. La opción seleccionada marcará el idioma de la página. La función que detectará la selección de la otra opción del select se llama:

```
cambiarIdioma(this.value)
```

El parámetro que se le pasa es el valor de la opción elegida, es para español y en para inglés. El pseudocódigo de esta función es el siguiente.

- Se guarda en una variable local la URL actual.
- Se guarda en una segunda variable la URL cortándola en "?", valor que indica el paso de parámetros.

- A la URL cortada se le suma la cadena "?iso=" más el valor del parámetro pasado a la función.
- En caso de que la página tuviese más parámetros se le suman.
- Se redirecciona a la URL resultado.

6.1.2.3 Página Índice

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.2.1 Interfaz index](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante. La estructura de la interfaz como código principal de esta página se encuentran en el archivo `index.php` situado en el directorio raíz de la web.

6.1.2.3.1 Código

La funcionalidad principal de esta página tiene como objetivo verificar la conexión del usuario. El código se ejecuta cuando el usuario pulsa el botón enviar del formulario de identificación. El pseudocódigo de esta función se describe a continuación.

- Se guarda en variables locales los valores introducidos para el correo y la contraseña en el formulario.
- Se establece conexión con la base de datos.
- Se busca en la base de datos una fila donde la columna correo sea igual al valor introducido como correo electrónico.
- Si se encuentra coincidencia.
 - Suma a la contraseña introducida por el usuario el valor del salt almacenado en la base de datos.
 - Calcula la función sha1 del resultado de la suma.
 - Si la contraseña almacenada en la base de datos es idéntica a la función resumen recién calculada
 - Inicializa las variables de sesión correo con el correo introducido y tipo con el tipo de usuario que tiene en la base de datos.
 - Redirecciona a la página panel.
 - Si no son iguales
 - Muestra un mensaje indicando que se han introducido datos incorrectos.
- Si no se encuentra coincidencia.
 - Muestra un mensaje indicando que se han introducido datos incorrectos.

6.1.2.3.2 Interfaz

En la ilustración 6.1 se puede observar el resultado final de la implementación de la interfaz diseñada en la ilustración 5.15 del punto [5.5.2.1 Interfaz index](#).

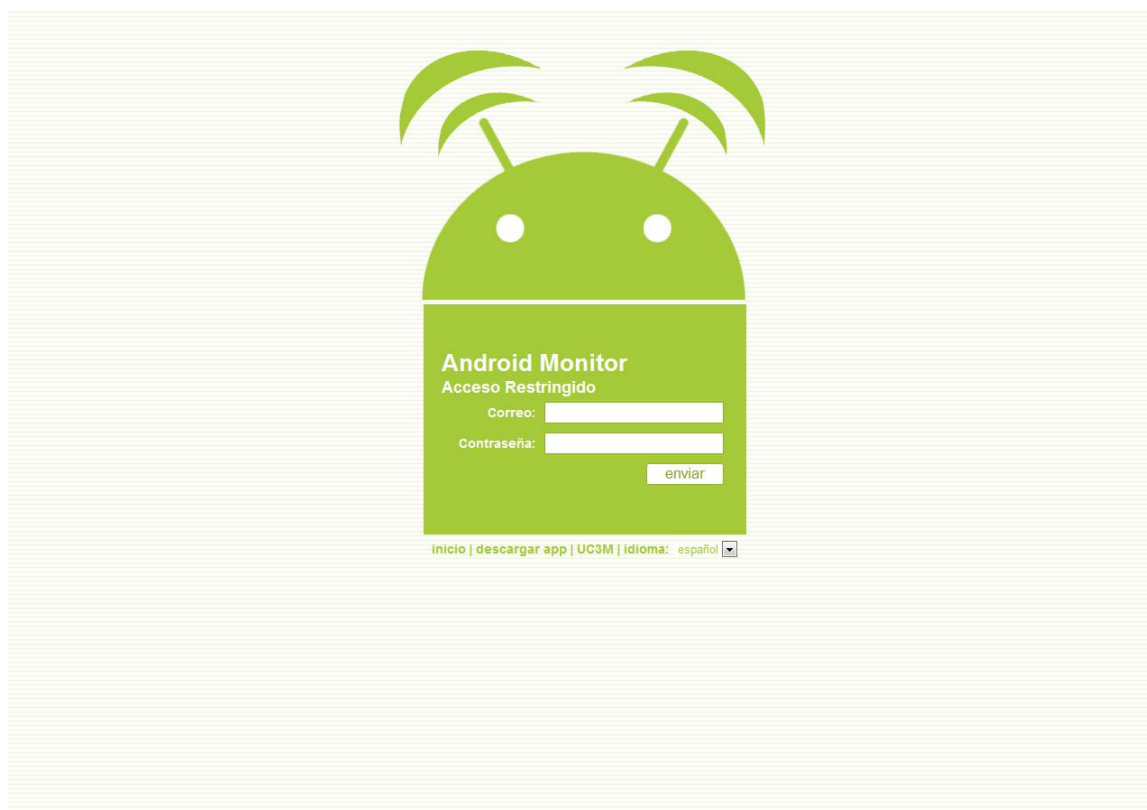


ILUSTRACIÓN 6.1: CAPTURA DE PANTALLA DE LA PÁGINA ÍNDIX DE LA WEB

6.1.2.4 Página descarga

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.2.2 Interfaz descargar app](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante. La estructura de la interfaz como código principal de esta página se encuentran en el archivo `descargar.php` situado en el directorio raíz de la web.

6.1.2.4.1 Código

El contenido principal de esta página consiste en una imagen que funciona como enlace. El enlace redirecciona al archivo `AndroidMonitor.apk` situado en la carpeta aplicación. La imagen consiste en un código QR que al ser leído por un lector de códigos redirecciona al mismo archivo. El código HTML de esto es el siguiente.

```
<a href="aplicacion/AndroidMonitor.apk"></a>
```

6.1.2.4.2 Interfaz

En la ilustración 6.2 se puede observar el resultado final de la implementación de la interfaz diseñada en la ilustración 5.16 del punto [5.5.2.2 Interfaz descargar app](#).

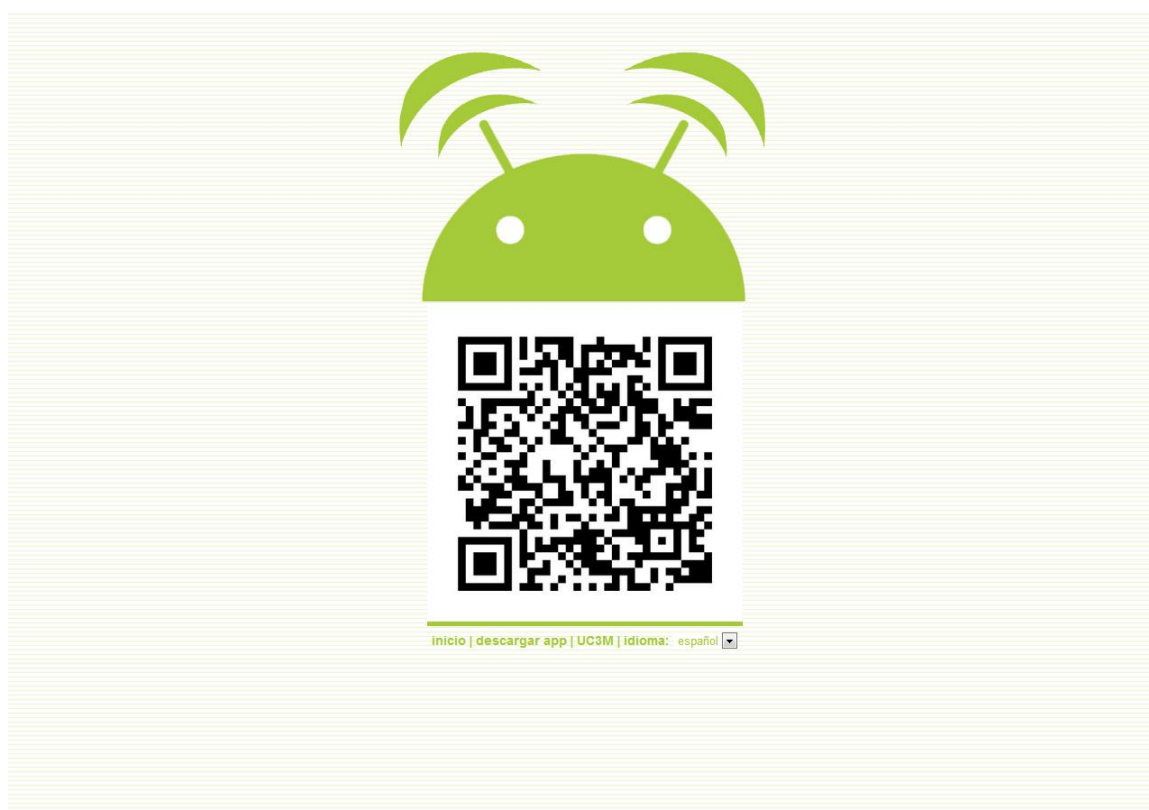


ILUSTRACIÓN 6.2: CAPTURA DE PANTALLA DE LA PÁGINA DESCARGA DE LA WEB

6.1.2.5 Página panel

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.2.3 Interfaz panel](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante. La estructura de la interfaz como código principal de esta página se encuentran en el archivo `panel.php` situado en el directorio raíz de la web.

6.1.2.5.1 Código

Los datos que se mostrarán en la página panel dependen del tipo de usuario que visite la página. Si el usuario es un usuario normal, sólo se mostrarán los dispositivos del usuario, si el usuario es un administrador, en la página se verán todos los dispositivos registrados. Por esto lo primero que se realizará será comprobar el tipo de usuario en la sesión. Una vez averiguado el tipo de usuario realizará las consultas necesarias para mostrar los datos en la pantalla. El pseudocódigo necesario para esto es el siguiente.

- Se establece la conexión con la base de datos.
- Si el usuario es de tipo administrador.
 - Se seleccionan todos los dispositivos almacenados en la tabla dispositivos de la base de datos.
- Si el usuario es de tipo usuario.
 - Se seleccionan todos los dispositivos almacenados en la tabla dispositivos de la base de datos que tengan como propietario el correo electrónico guardado en la variable de sesión.
- Mientras se lea una fila en la consulta realizada.
 - Se crea un script JavaScript

Cada script creado estará compuesto por tres funciones, el pseudocódigo de este script es el siguiente.

- Se crea una variable local con la dirección al archivo actualización.php situado en el directorio php_code pasándole como parámetro la iso actual.
- Ejecuta la función actualización_reloj cada quinientos milisegundos.
- Si el navegador es Internet Explorer se crea un objeto de tipo ActiveXObject para recibir información en HTML.
- Si el navegador no es Internet Explorer se crea un objeto de tipo XMLHttpRequest para recibir información en HTML.
- Función ReqChange()
 - Si se ha recibido la información correctamente.
 - Si la información recibida es válida.
 - Se obtiene el contenido.
 - Se imprime el contenido recibido en el div con identificador listadispositivos.
 - Si la información recibida no es válida.
 - Se imprime un mensaje de error en el div con identificador listadispositivos.
- Función refrescarDatos()
 - Se obtienen los datos del resultado de la ejecución del código actualización.php con el objeto XMLHttpRequest creado.

- o Se ejecuta la función `ReqChange` con el objeto `XMLHttpRequest` creado.
 - o Se envía un `null` con el objeto `XMLHttpRequest` creado.
- Función `actualización_reloj()`
 - o Invoca a la función `refrescarDatos`

El archivo `actualización.php` que se invoca en el script anterior se sitúa en el directorio `php_code/` Su principal objetivo es realizar las consultas a la base de datos e imprimir el código HTML necesario para mostrar el contenido solicitado. El pseudocódigo de esta función es el siguiente.

- Se establece la conexión con la base de datos.
- Si el usuario es de tipo administrador.
 - o Se seleccionan todos los dispositivos almacenados en la tabla `dispositivos` de la base de datos ordenándolos por su estado en orden descendiente.
- Si el usuario es de tipo usuario.
 - o Se seleccionan todos los dispositivos almacenados en la tabla `dispositivos` de la base de datos que tengan como propietario el correo electrónico guardado en la variable de sesión.
- Mientras se lea una fila en la consulta realizada.
 - o Se abre un contenedor.
 - o Se escribe el HTML de la primera fila que incluye el correo del propietario, el modelo del teléfono, la fecha del último registro, el enlace a los detalles y el enlace al histórico de la consulta.
 - o Se escribe el título de la columna periodo.
 - o Se escribe en los datos de la columna periodo.
 - o Se escribe el título de la columna acelerómetro
 - o Se busca en la tabla de datos del sensor acelerómetro la fila que tenga como dispositivo el `id` de la consulta y como fecha la fecha del último registro de la consulta.
 - o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
 - o Si no se encuentran resultados
 - Se escribe un `'-'`.
 - o Se escribe el título de la columna gravedad
 - o Se busca en la tabla de datos del sensor gravedad la fila que tenga como dispositivo el `id` de la consulta y como fecha la fecha del último registro de la consulta.
 - o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
 - o Si no se encuentran resultados
 - Se escribe un `'-'`.
 - o Se escribe el título de la columna giroscopio

- o Se busca en la tabla de datos del sensor giroscopio la fila que tenga como dispositivo el id de la consulta y como fecha la fecha del último registro de la consulta.
- o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
- o Si no se encuentran resultados
 - Se escribe un '-'.
- o Se escribe el título de la columna sensor de luz
- o Se busca en la tabla de datos del sensor de luz la fila que tenga como dispositivo el id de la consulta y como fecha la fecha del último registro de la consulta.
- o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
- o Si no se encuentran resultados
 - Se escribe un '-'.
- o Se escribe el título de la columna aceleración lineal
- o Se busca en la tabla de datos del sensor de aceleración lineal la fila que tenga como dispositivo el id de la consulta y como fecha la fecha del último registro de la consulta.
- o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
- o Si no se encuentran resultados
 - Se escribe un '-'.
- o Se escribe el título de la columna campo magnético
- o Se busca en la tabla de datos del sensor de campo magnético la fila que tenga como dispositivo el id de la consulta y como fecha la fecha del último registro de la consulta.
- o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
- o Si no se encuentran resultados
 - Se escribe un '-'.
- o Se escribe el título de la columna orientación
- o Se busca en la tabla de datos del sensor de orientación la fila que tenga como dispositivo el id de la consulta y como fecha la fecha del último registro de la consulta.
- o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
- o Si no se encuentran resultados
 - Se escribe un '-'.
- o Se escribe el título de la columna presión
- o Se busca en la tabla de datos del sensor de presión la fila que tenga como dispositivo el id de la consulta y como fecha la fecha del último registro de la consulta.

- o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
- o Si no se encuentran resultados
 - Se escribe un '-\.
- o Se escribe el título de la columna proximidad
- o Se busca en la tabla de datos del sensor de proximidad la fila que tenga como dispositivo el id de la consulta y como fecha la fecha del último registro de la consulta.
- o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
- o Si no se encuentran resultados
 - Se escribe un '-\.
- o Se escribe el título de la columna vector de rotación
- o Se busca en la tabla de datos del sensor de vector de rotación la fila que tenga como dispositivo el id de la consulta y como fecha la fecha del último registro de la consulta.
- o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
- o Si no se encuentran resultados
 - Se escribe un '-\.
- o Se escribe el título de la columna temperatura
- o Se busca en la tabla de datos del sensor de temperatura la fila que tenga como dispositivo el id de la consulta y como fecha la fecha del último registro de la consulta.
- o Si se encuentran resultados
 - Se escriben los datos de la fila encontrada.
- o Si no se encuentran resultados
 - Se escribe un '-\.
- o Se cierra el contenedor.

6.1.2.5.2 Interfaz

En la ilustración 6.3 se puede observar el resultado final de la implementación de la interfaz diseñada en la ilustración 5.17 del punto [5.5.2.3 Interfaz panel](#).

antonioblopez@gmail.com, Galaxy Nexus, 2012-08-28 21:47:43					
Period	Accelerometer	Gravity	Gyroscope	Light Sensor	Linear acceleration
1	x: -6.799 m/s ² y: -5.781 m/s ² z: 2.188 m/s ²	x: -6.854 m/s ² y: -6.793 m/s ² z: 1.762 m/s ²	x: 0 m/s ² y: 0 m/s ² z: -1 m/s ²	4.143 lux	x: 0.056 m/s ² y: 1.012 m/s ² z: 0.426 m/s ²
	Magnetic field x: 13.100 μ T y: 29.800 μ T z: -29.600 μ T	Orientation x: 72.693 ° y: 43.828 ° z: -75.584 °	Pressure 934.640 mB	Proximity 5 cm	Rotation Vector x: 0.099 m/s ² y: 0.633 m/s ² z: -0.619 m/s ²

antonioblopez@gmail.com, Nexus One, 2012-08-28 20:13:10					
Period	Accelerometer	Gravity	Gyroscope	Light Sensor	Linear acceleration
1	x: -2.792 m/s ² y: 4.100 m/s ² z: 7.736 m/s ²	x: -2.903 m/s ² y: 3.907 m/s ² z: 8.064 m/s ²	-	10240.000 lux	x: 0.111 m/s ² y: 0.193 m/s ² z: -0.328 m/s ²
	Magnetic field x: 14.562 μ T y: -9.875 μ T z: -15.438 μ T	Orientation x: 264.000 ° y: -26.000 ° z: -18.000 °	Pressure -	Proximity 9 cm	Rotation Vector x: 0.034 m/s ² y: 0.266 m/s ² z: 0.694 m/s ²

cuatro@cuatro.com, Nexus One, 2012-08-27 21:06:16					
Period	Accelerometer	Gravity	Gyroscope	Light Sensor	Linear acceleration
1	x: -0.531 m/s ² y: 3.214 m/s ² z: 9.997 m/s ²	x: -1.429 m/s ² y: 2.608 m/s ² z: 9.011 m/s ²	-	225.000 lux	x: 0.898 m/s ² y: 0.606 m/s ² z: 0.986 m/s ²
	Magnetic field x: -3.250 μ T y: 16.875 μ T z: -47.125 μ T	Orientation x: 12.000 ° y: -18.000 ° z: -3.000 °	Pressure -	Proximity 9 cm	Rotation Vector x: 0.150 m/s ² y: 0.051 m/s ² z: -0.168 m/s ²

dos@dos.com, Nexus One, 2012-08-17 19:40:12					
Period	Accelerometer	Gravity	Gyroscope	Light Sensor	Linear acceleration
2	x: 1.035 m/s ² y: -0.004 m/s ² z: -0.004 m/s ²	x: 1.039 m/s ² y: -0.004 m/s ² z: -0.004 m/s ²	-	10.000 lux	x: -0.004 m/s ² y: -0.004 m/s ² z: -0.004 m/s ²

ILUSTRACIÓN 6.3: CAPTURA DE PANTALLA DE LA PÁGINA PANEL DE LA WEB

6.1.2.6 Página detalles

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.2.4 Interfaz detalles](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante. Tanto la estructura de la interfaz como código principal de esta página se encuentran en el archivo `monitorizarDispositivo.php` situado en el directorio raíz de la web.

6.1.2.6.1 Código

El código de esta página es similar al código de la página Panel. La principal diferencia recae en que a esta página se le deben pasar dos parámetros por URL y sólo muestra la monitorización de un dispositivo identificado por los parámetros pasados.

Como ya se ha mencionado el primer código que se ejecutará en esta página se encargará de tomar los parámetros que se le han enviado. El pseudocódigo es el siguiente.

- Se establece conexión con la base de datos.
- Si existe el parámetro 'correo'
 - o Se guarda en una variable local el valor del parámetro.
- Si no se encuentra el parámetro.

- o Se crea la variable local con contenido vacío.
- Si existe el parámetro 'modelo'
 - o Se guarda en una variable local el valor del parámetro.
- Si no se encuentra el parámetro.
 - o Se crea la variable local con contenido vacío.
- Si la variable que guarda el parámetro correo o la variable que guarda el parámetro modelo se encuentran vacías.
 - o Se redirecciona la página a la página panel.
- Se busca en la base de datos el dispositivo que tenga como propietario el correo pasado como parámetro y como modelo el modelo pasado como parámetro.
- Si no se ha encontrado ningún resultado.
 - o Se redirecciona la página a la página panel.
- Si se ha encontrado algún resultado.
 - o Se escribe el script

El Script que se escribe es idéntico al script definido en el punto [6.1.2.5.1 Código](#). La principal diferencia se encuentra en que este script ejecuta el código que se encuentra en el archivo `datosSensores.php` y el refresco de los datos está definido por el periodo del dispositivo.

6.1.2.6.2 Interfaz

En la ilustración 6.4 se puede observar el resultado final de la implementación de la interfaz diseñada en la ilustración 5.18 del [punto 5.5.2.4 Interfaz detalles](#).



ILUSTRACIÓN 6.4: CAPTURA DE PANTALLA DE LA PÁGINA DETALLES DE LA WEB

6.1.2.7 Página histórico

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.2.5 Interfaz histórico](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante. La estructura de la interfaz como código principal de esta página se encuentran en el archivo `historico.php` situado en el directorio raíz de la web.

6.1.2.7.1 Código

La función principal de esta página consiste en validar el formulario de consulta de históricos y llamar a las funciones que muestren los datos según los parámetros introducidos.

A esta página se puede acceder mediante el enlace de la barra superior de la web, o desde los enlaces de las tablas de datos de los dispositivos de la página panel. Desde esta segunda opción se accede a la página pasándole como parámetros el correo y el modelo del dispositivo seleccionado. Lo primero que hará la página será comprobar si se le han pasado parámetros por URL. Si es así los selectores de Usuario y dispositivo estarán inicializados a estos valores. Si no se le han pasado parámetros a la página estos selectores están en una opción nula.

Para poder seleccionar un dispositivo se deberá tener seleccionado primero un usuario. El contenido del selector de dispositivos se rellenará mediante un código que se ejecutará cuando se detecte que se ha seleccionado una opción en el selector de usuarios. Esta función llamará al código situado en el archivo `selectDispositivo.php` que se sitúa en el directorio `php_code/` pasándole como parámetro el usuario seleccionado. `selectDispositivo` Ejecutará una consulta a la base de datos buscando los dispositivos que tengan asociados el correo pasado como parámetro, y escribirá los resultados encontrados en el selector de dispositivos.

Al pulsar el botón enviar del formulario lo primero que realizará el código será comprobar que se han rellenado todos los campos disponibles, en caso de que algún campo no esté rellenado el código mostrará un mensaje de alerta situando el cursor sobre el campo que ha generado la alerta. Si todos los datos han sido introducido se procederá a comprobar que el intervalo de fechas ha sido bien introducido siendo la fecha y hora “de” menor que la fecha y hora “a”. Finalmente se enviarán estos datos al documento `consultaHistorico.php` que realizará las consultas necesarias para mostrar los datos solicitados. Mediante una función Ajax se cargará el resultado en pantalla debajo del formulario.

6.1.2.7.2 Interfaz

En la ilustración 6.5 se puede observar el resultado final de la implementación de la interfaz diseñada en la ilustración 5.19 del punto [5.5.2.5 Interfaz histórico](#).

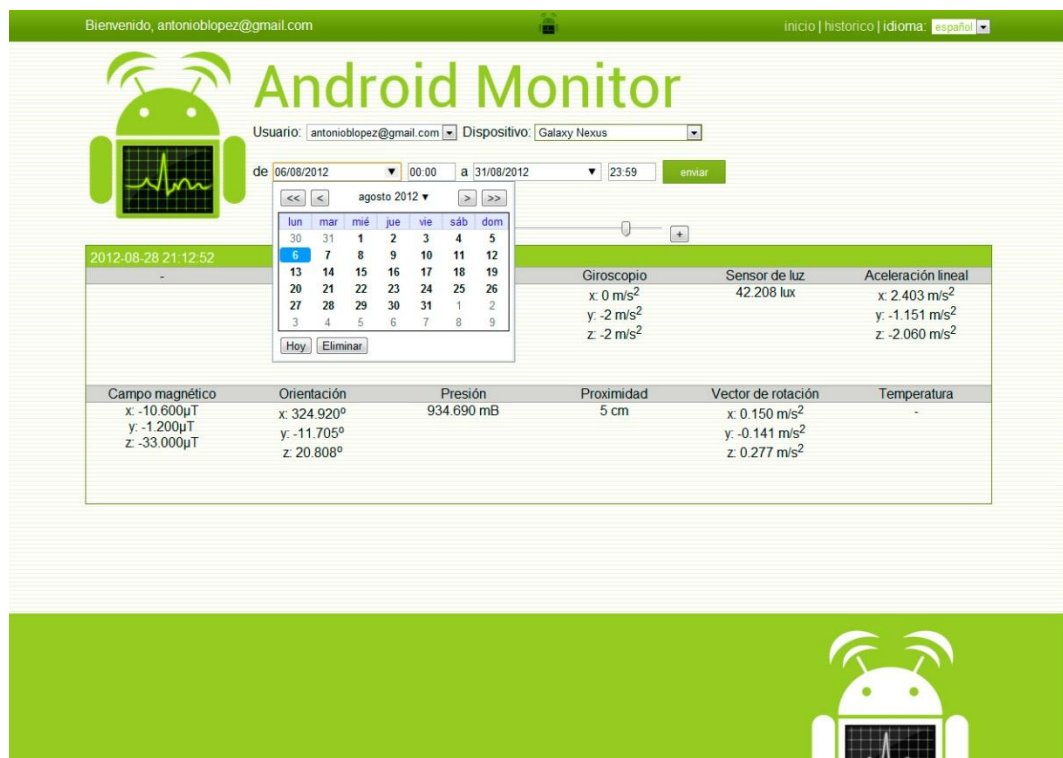


ILUSTRACIÓN 6.5: CAPTURA DE PANTALLA DE LA PÁGINA HISTÓRICO DE LA WEB

6.1.2.8 Navegación

Siguiendo las directrices del punto [5.5.2 Servidor web](#) del apartado de Diseño, se han implementado la navegación entre las interfaces de usuario de la página Web. En la ilustración 6.6 se puede apreciar la navegación entre las diferentes interfaces definidas en los puntos anteriores.

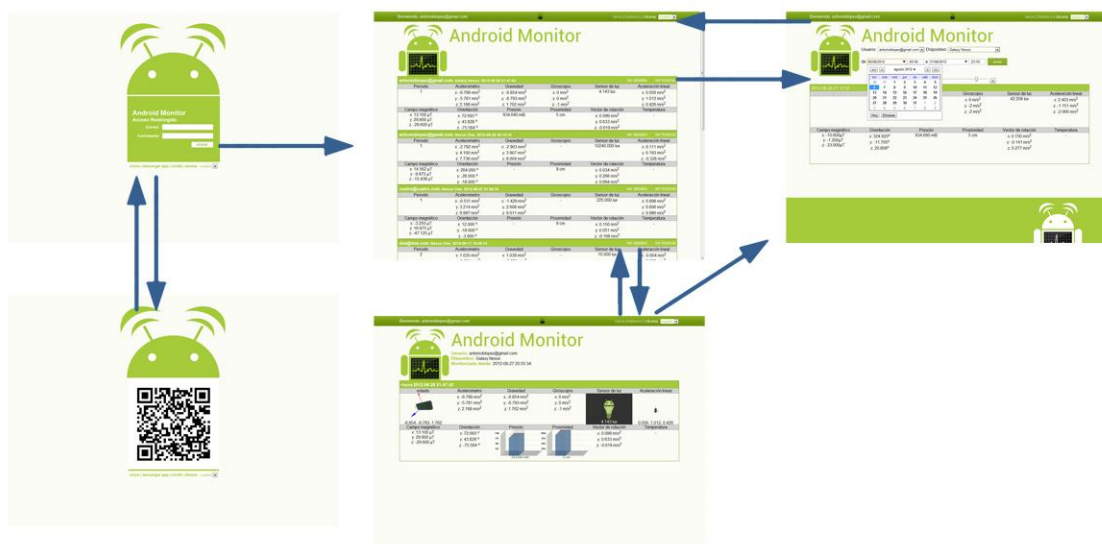


ILUSTRACIÓN 6.6: NAVEGACIÓN DE PANTALLAS EN LA PÁGINA WEB

6.2 Aplicación móvil

En este apartado se explicarán los principales detalles del desarrollo de la aplicación para dispositivos móviles. Se dividirá el desarrollo de la aplicación en dos partes: el cliente web y la aplicación. El cliente web será la parte que realice las comunicaciones con el servidor. La aplicación se considerará a parte del programa con la que interactuará el usuario en su dispositivo.

6.2.1 Cliente Web

En este apartado se completará el desarrollo de las funciones que se comenzaron a desarrollar en el apartado 4.4.1.1, en dicho apartado se desarrollo la parte relacionada con el servidor web, en este apartado se procederá a desarrollar la parte que concierne a la aplicación móvil.

Todas las conexiones con el servidor se realizarán en hilos secundarios con la clase AsyncTask. El principal motivo de realizarlo de esta manera reside en que desde la versión 4 de Android no se permite a las aplicaciones que realicen conexiones con servidores externos en el hilo principal de la aplicación, por lo que para que la aplicación

desarrollada sea compatible con las actuales versiones del sistema operativo Android, como ya se ha comentado, todas las conexiones se harán en segundo plano.

Los AsyncTask se definen con la siguiente línea:

```
AsyncTask<Params, Progress, Result> nombre = new
AsyncTask<Params, Progress, Result>() {
```

Dentro de esta definición se crea la función:

```
protected String doInBackground(Void... params)
```

en cuyo interior se debe escribir el código que se quiere ejecutar en segundo plano. Una vez terminada la definición del AsyncTask se ejecuta con la función `execute()` ;

6.2.1.1 Registro

El código que realiza la función de registro por parte de la aplicación móvil se encuentra en la clase `MainActivity.java` que se encuentra en el directorio `/src/es/uc3m/androidmonitor`.

La función se llama `String enviarRegistro(JSONObject js)` es invocada desde la función `onClick(View v)` que se ejecuta cuando el usuario pulsa sobre el botón enviar de la interfaz.

Como se puede leer, a la función `enviarRegistro` se le pasa como parámetro un objeto `JSONObject`, este objeto se crea en la función `onClick`, se compone de una estructura formada por los elementos descritos en el punto [5.3.1 Registro](#), cada elemento contará con una etiqueta o identificador y un valor.

La función `enviarRegistro` lo primero que hará será definir el AsyncTask necesario para ejecutar la comunicación con el servidor. Este AsyncTask devolverá una cadena que será la respuesta recibida por el servidor. El pseudocódigo realizado en el AsyncTask es el siguiente.

- Se crea un objeto de tipo `HttpParams`.
- Se establece el tiempo de espera de la conexión.
- Se establece el tiempo de espera del socket predeterminado.
- Se crea un objeto de tipo `HttpClient` para el objeto `HttpParams` creando anteriormente.
- Se establece la URL al archivo de destino `registro.php` del directorio `php_code` del servidor remoto.
- Se crea una solicitud como un objeto `HttpPost` con la URL creada en el paso anterior.
- Se establece la entidad de la solicitud anterior con los bytes del parámetro que ha recibido la función.

- Se establece la cabecera de la solicitud dando como nombre "json" y como valor el parámetro que ha recibido la función en forma de cadena de texto.
- Se ejecuta la función `execute` del objeto `HttpClient` pasándole como parámetro la solicitud y se guarda el resultado en un objeto `HttpResponse`.
- Se guarda en un objeto `HttpEntity` la entidad del objeto `HttpResponse`.
- Si la entidad no es nula.
 - o Se toma el contenido de la entidad.
 - o Se convierte a cadena de texto.
 - o Se retorna la cadena.

Una vez definido el contenido del `AsyncTask` se ejecuta y se guarda el resultado de su ejecución en una variable que es devuelta por el método.

6.2.1.1.1 Interfaz

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.3.1 Registro](#) de la sección Diseño. La estructura de la interfaz se define en el archivo `signup.xml` situado en la carpeta `res/layout`. En la ilustración 6.7 se puede observar el resultado final de la implementación de esta interfaz.

La imagen muestra una interfaz de usuario para el registro en una aplicación llamada AndroidMonitor. El fondo es de color verde lima. En la parte superior hay un icono blanco de la cabeza de un robot Android con antenas. Debajo del icono, el texto dice: "Introduzca los siguientes datos para registrarse en AndroidMonitor, podrá usarlos para acceder a la aplicación via web". Hay tres campos de entrada de texto: el primero está etiquetado "correo" y contiene el texto "correo"; el segundo está etiquetado "contraseña" y contiene el texto "contraseña"; el tercero está etiquetado "periodo (s)" y está vacío. Debajo del tercer campo, hay un texto pequeño que dice "*periodo de envío de datos al servidor". En la parte inferior derecha hay un botón gris con el texto "enviar".

ILUSTRACIÓN 6.7: CAPTURA DE PANTALLA DE LA INTERFAZ REGISTRO DE LA APLICACIÓN

6.2.1.2 Cambio de periodo

El código que realiza la función de cambiar periodo por parte de la aplicación móvil se encuentra en la clase `Opciones.java` que se encuentra en el directorio `/src/es/uc3m/androidmonitor`. El método que realiza esta función se llama `public void cambiarPeriodo()` es invocado por el método `public void onClick(View v)` cuando detecta que se ha pulsado el botón asociado al cambio de periodo. El pseudocódigo de la función `cambiarPeriodo()` es el siguiente.

- Se crea la conexión con la base de datos.
- Se guarda en una variable local el dato introducido en el input de periodo de la vista.
- Si el periodo introducido es menor que 1
 - se iguala la variable a 1.
 - Se imprimir un mensaje advirtiéndole de que el periodo no puede ser inferior a 1.
- Se selecciona de la base de datos los datos correo y modelo de la tabla Usuario.
- Si el dispositivo tiene conexión a Internet.
 - Se crea un `AsyncTask` similar al descrito en el apartado 4.4.2.1.1 Registro. Este `AsyncTask` tendrá como archivo destino el fichero llamado `cambiarPeriodo.php`.
 - Lo primero que realizará el `AsyncTask` será crear el objeto `JSONObject` teniendo como parámetros el correo y modelo extraídos de la base de datos y el tiempo nuevo introducido por el usuario.
 - Se ejecuta el `AsyncTask`.
 - Si la cadena devuelta contiene un OK
 - Se actualiza el valor del periodo guardado en la tabla Config de la base de datos al nuevo valor introducido.
 - Se invoca un mensaje toast indicando al usuario de que se ha realizado el cambio correctamente.
 - Si no
 - Se invoca un mensaje toast indicando al usuario de que se ha producido un error.
- Si el dispositivo no tiene conexión a Internet.
 - Se crea un mensaje con formato json pasando como parámetros el correo y el modelo sacados de la base de datos y el dato introducido como periodo.
 - Se guarda en la tabla MensajesPendientes de la base de datos el mensaje creado con el tipo con valor 1.
 - Se actualiza el valor del periodo guardado en la tabla Config de la base de datos al nuevo valor introducido.

- Se cierran las conexiones a la base de datos.

6.2.1.3 Cambio de contraseña

El código que realiza la función de cambiar contraseña por parte de la aplicación móvil se encuentra en la clase `Opciones.java` que se encuentra en el directorio `/src/es/uc3m/androidmonitor`. El método que realiza esta función se llama `public void cambiarPass()` es invocado por el método `public void onClick(View v)` cuando detecta que se ha pulsado el botón asociado al cambio de contraseña de la vista de configuración. El pseudocódigo de la función `cambiarPass()` es el siguiente.

- Si el dispositivo tiene conexión a Internet.
 - o Se crea la conexión con la base de datos.
 - o Se guarda en una variable local los datos introducidos en el formulario de cambio de contraseña de la interfaz: la actual contraseña y la nueva contraseña.
 - o Se selecciona de la base de datos los datos correo, contraseña y salt de la tabla Usuario.
 - o Se realiza la función hash de la contraseña actual introducida por el usuario con el salt extraído de la base de datos.
 - o Si el resultado de la función hash es igual a la contraseña almacenada en la base de datos
 - Se calcula un nuevo salt aleatorio.
 - Se realiza la función hash de la nueva contraseña introducida por el usuario con el nuevo salt calculado.
 - Se crea un `AsyncTask` similar al descrito en el apartado 4.4.2.1.1 Registro. Este `AsyncTask` tendrá como archivo destino el fichero llamado `cambiarPass.php`.
 - Lo primero que realizará el `AsyncTask` será crear el objeto `JSONObject` teniendo como parámetros el correo y la contraseña extraídas de la base de datos, el nuevo salt calculado y el resultado de la función resumen de la nueva contraseña con el nuevo salt.
 - Se ejecuta el `AsyncTask`.
 - Si la cadena devuelta contiene un OK
 - Se actualiza el valor de la contraseña y el salt guardados en la tabla Usuario de la base de datos a los nuevos valores calculados.

- Se invoca un mensaje toast indicando al usuario de que se ha realizado el cambio correctamente.
- Si no
 - Se invoca un mensaje toast indicando al usuario de que se ha producido un error.
- o Si no coinciden
 - Se invoca un mensaje toast indicando al usuario que ha introducido incorrectamente la contraseña actual.
- Si el dispositivo no tiene conexión a Internet.
 - o Se invoca un mensaje toast indicando al usuario que es necesario la conexión a Internet para cambiar la contraseña.
- Se cierran las conexiones a la base de datos.

6.2.1.4 Monitorización

El código que realiza la función de monitorización por parte de la aplicación móvil se encuentra en la clase `MonitorService.java` que se encuentra en el directorio `/src/es/uc3m/androidmonitor`. El servicio puede ser inicializado de tres formas distintas.

Al iniciar por primera vez la aplicación:

Nada más terminar el registro en la aplicación al abrirla por primera vez, antes de que la aplicación lance el Activity del menú, la aplicación iniciará el servicio.

Por la función `public void monitorizar()`:

El usuario tiene la capacidad de activar y desactivar la monitorización del sistema, esta función otorga al usuario esta capacidad. Se verá en más detalle el funcionamiento de este método en el punto [6.2.2.5 Configuración](#).

Al encenderse el dispositivo móvil:

Una de las características más importantes de un servicio es la capacidad de comenzar a ejecutarse sin necesidad de que interactúe el usuario. Para que una aplicación pueda ejecutar un servicio al iniciarse el sistema operativo se le debe otorgar el permiso:

```
android.permission.RECEIVE_BOOT_COMPLETED
```

Se creará una clase llamada `public class iniciador` que extenderá de la clase `BroadcastReceiver`. En el archivo `AndroidManifest` se deberá declarar esta clase de la siguiente forma.

```
<receiver android:name=".iniciador">
    <intent-filter>
        <action
            android:name="android.intent.action.BOOT_COMPLETED"/>
    </intent-filter>
</receiver>
```

Con estas líneas se indica que el receiver está esperando un mensaje de tipo `BOOT_COMPLETED`. Cuando reciba este mensaje se ejecutará el código de la clase `iniciador`, que simplemente iniciará el servicio de la aplicación en su método `public void onReceive(Context context, Intent intent)`.

El algoritmo del servicio está definido por el diagrama de flujo del punto [5.4 Aplicación Servicio](#) la clase `MonitorService` al ser un servicio extiende de la clase `Service` y para el control de los sensores del dispositivo debe implementar la clase `SensorEventListener`.

```
public class MonitorService extends Service implements
    SensorEventListener
```

Los servicios al igual que los `BroadcastReceiver` deben ser declarados en el archivo `AndroidManifest`, la declaración es la siguiente.

```
<service android:name=".MonitorService">
    <intent-filter>
        <action android:name="MonitorService"/></action>
    </intent-filter>
</service>
```

Con esto la clase `MonitorService` ya funcionará como servicio de la aplicación. La clase `MonitorService` está formada principalmente por tres funciones: `onCreate`, `onStart` y `onSensorChanged`. A continuación se describirán los objetivos principales de estas funciones.

- **`public void onCreate()`**

El primer método que se ejecuta al crear el servicio. El objetivo de este método es inicializar todos los datos necesarios para ejecutar el servicio, inicializando variables y sacando de la base de datos los datos necesarios. Una vez inicializado las variables registrará los sensores del dispositivo y e iniciará para cada uno de ellos la escucha de eventos que se ejecutará en la clase `onSensorChanged(SensorEvent event)`. Para declarar los sensores se deberá hacer uso de la clase `SensorManager`, esta clase permite acceder a los sensores del dispositivo.

- **`public void onStart(final Intent intent, final int startId)`**

La función se ejecuta después de la función `onCreate`. La función crea un objeto de tipo `Runnable` que ejecutará un código cada cierto periodo. Esta es la función que se encargará de enviar al servidor los mensajes. Primero detectará si tiene conexión a Internet, si no tiene conexión guardará los mensajes en la base de datos. Si sí se tiene conexión comprobará si han mensajes pendientes de enviar. Si hay mensajes pendientes creará un hilo `AsynkTask` que se encargará de enviar estos mensajes. A continuación creará un hilo que se encargará de enviar los datos almacenados en las variables globales que contienen los datos de los sensores al servidor.

El periodo de ejecución de la función `Runnable` viene dado por el periodo que introduce el usuario, para que se cumpla este periodo se calculará el tiempo que tarda en ejecutarse la función, se tomará el tiempo actual al empezar y al terminar, la resta de estos dos dará como resultado el tiempo de ejecución. Al periodo introducido por el usuario se le restará el tiempo de ejecución y ese tiempo será el que marcará cuando se volverá a ejecutar la función. De esta manera se cumplirá la mayoría de las veces el periodo impuesto por el usuario.

- **`public void onSensorChanged(SensorEvent event)`**

La función se ejecuta cuando se produce un cambio en el valor de un sensor. El método se encarga de averiguar que sensor ha cambiado sus valores y de actualizar las variables globales que guardan los valores de dicho sensor.

6.2.2 Aplicación

A continuación se describirán las principales funcionalidades de la aplicación que interactúen con el usuario. Se dividirán por interfaces para su mejor ubicación.

6.2.2.1 Identificación

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.1.2 Identificación](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante.

6.2.2.1.1 Código

La identificación de usuario se lleva a cabo en la clase MainActivity.java java que se encuentra en el directorio `/src/es/uc3m/androidmonitor`. El código se sitúa en el método `public void onClick(View v)`, que es llamado cuando se detecta que se ha pulsado sobre el botón aceptar de la interfaz de usuario. El pseudocódigo del método es similar al descrito en el punto [6.1.2.3.1 Código](#), donde se describe la función identificar de la página web. Si la identificación se ha realizado correctamente el usuario será redirigido a la página menú.

6.2.2.1.2 Interfaz

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.1.2 Identificación](#) de la sección Diseño. La estructura de la interfaz se define en el archivo `login.xml` situado en la carpeta `res/layout`. En la ilustración 6.8 se puede observar el resultado final de la implementación de esta interfaz.



ILUSTRACIÓN 6.8: CAPTURA DE PANTALLA DE LA INTERFAZ IDENTIFICACIÓN DE LA APLICACIÓN

6.2.2.2 Menú

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.2 Interfaz menú](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante.

6.2.2.2.1 Código

El código de esta página se sitúa en el archivo `MenuPrincipalActivity.java` que se encuentra en el directorio `/src/es/uc3m/androidmonitor`. La única funcionalidad con la que cuenta esta página trata de escuchar los botones que se encuentran en ella, identificar cual ha sido presionado y redireccionar al usuario a la interfaz correspondiente.

6.2.2.2.2 Interfaz

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.2 Interfaz menú](#) de la sección Diseño. La estructura de la interfaz se define en el archivo `activity_main.xml` situado en la carpeta `res/layout`. En la ilustración 6.9 se puede observar el resultado final de la implementación de esta interfaz.

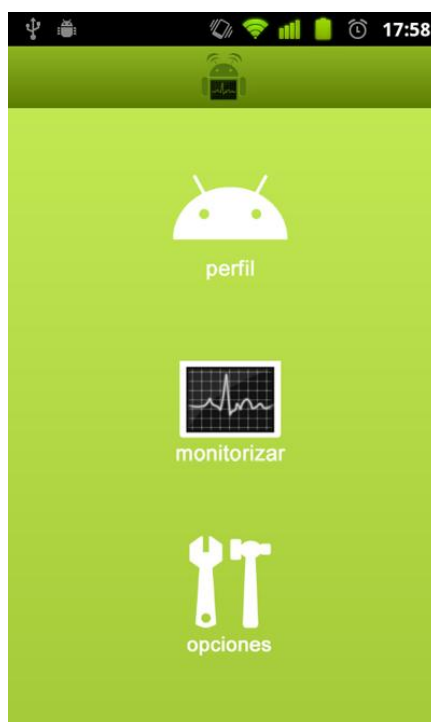


ILUSTRACIÓN 6.9: CAPTURA DE PANTALLA DE LA INTERFAZ MENÚ DE LA APLICACIÓN

6.2.2.3 Perfil

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.3 Interfaz perfil](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante.

6.2.2.3.1 Código

El código de la página situado en el archivo `Perfil.java`, archivo que se encuentra en el directorio `/src/es/uc3m/androidmonitor`, realiza consultas a la base de datos de los datos personales del usuario y los imprime en la interfaz que es mostrada al usuario. El código también escucha el botón situado en la esquina superior izquierda, si detecta que es pulsado redirecciona al usuario a la interfaz menú.

6.2.2.3.2 Interfaz

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.3 Interfaz perfil](#) de la sección Diseño. La estructura de la interfaz se define en el archivo `activity_perfil.xml` situado en la carpeta `res/layout`. En la ilustración 6.10 se puede observar el resultado final de la implementación de esta interfaz.

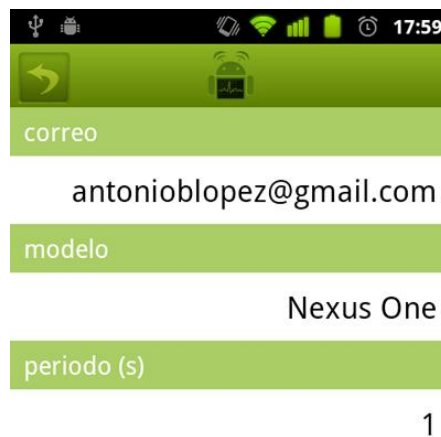


ILUSTRACIÓN 6.10: CAPTURA DE PANTALLA DE LA INTERFAZ PERFIL DE LA APLICACIÓN

6.2.2.4 Monitorización

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.4 Interfaz monitorización](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante.

6.2.2.4.1 Código

La funcionalidad de esta página se encuentra en el archivo `Monitor.java` situado en el directorio `/src/es/uc3m/androidmonitor`. El código lee los valores de los sensores almacenados en las variables globales del servicio de la aplicación y los imprime en pantalla. Al igual que la interfaz perfil también incluye un código para reaccionar a la pulsación del botón volver.

6.2.2.4.2 Interfaz

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.4 Interfaz monitorización](#) de la sección Diseño. La estructura de la interfaz se define en el archivo `activity_monitor.xml` situado en la carpeta `res/layout`. En la ilustración 6.11 se puede observar el resultado final de la implementación de esta interfaz.

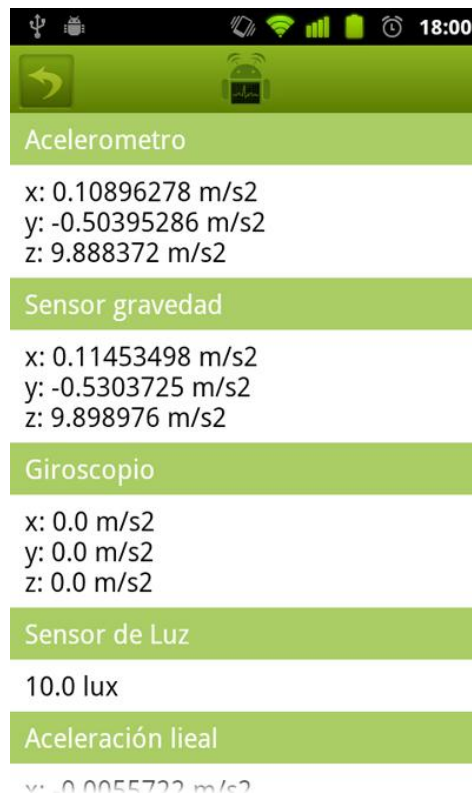


ILUSTRACIÓN 6.11: CAPTURA DE PANTALLA DE LA INTERFAZ MONITOR DE LA APLICACIÓN

6.2.2.5 Configuración

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.5 Interfaz configuración](#) de la sección Diseño. A continuación se describirá el código necesario para implementar las funcionalidades de esta página y la interfaz resultante.

6.2.2.5.1 Código

El código de esta página se incluye en el archivo `Opciones.java` del directorio `/src/es/uc3m/androidmonitor`. A parte de las funcionalidades cambiar contraseña y cambiar periodo que ya han sido implementadas en los puntos [5.3.3 Cambio de contraseña](#) y [5.3.2 Cambio de periodo](#), respectivamente, la página cuenta con una tercera función llamada `public void monitorizar()` que se ejecuta cuando el usuario pulsa sobre el botón que activa o desactiva la monitorización. El pseudocódigo de esta función es el siguiente:

Monitorizar()

- Se guarda en una variable local el estado del botón.
- Se crea conexión con la base de datos.
- Si el botón está activado
 - Se actualiza el estado de la monitorización en la tabla config de la base de datos.
 - Se crea un nuevo servicio `MonitorService`.
 - Se inicializa el servicio.
 - Se imprime un mensaje toast que informa al usuario que el dispositivo se está monitorizando.
- Si el botón está desactivado.
 - Se actualiza el estado de la monitorización en la tabla config de la base de datos.
 - Se detiene el servicio `MonitorService`.
 - Se imprime un mensaje toast que informa al usuario que la monitorización se ha desactivado.
- Se cierran las conexiones con la base de datos.

6.2.2.5.2 Interfaz

El diseño y descripción de elementos que debe mostrar esta página se encuentra en el punto [5.5.1.5 Interfaz configuración](#) de la sección Diseño. La estructura de la interfaz se define en el archivo `activity_opciones.xml` situado en la carpeta `res/layout`. En la ilustración 6.12 se puede observar el resultado final de la implementación de esta interfaz.

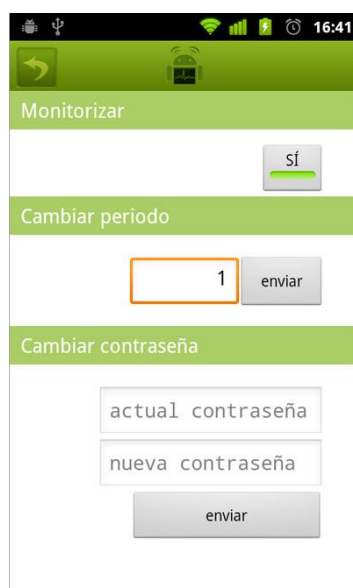


ILUSTRACIÓN 6.12: CAPTURA DE PANTALLA DE LA INTERFAZ CONFIGURACIÓN DE LA APLICACIÓN

6.2.2.6 Navegación

Siguiendo las directrices del punto [5.5.1 Aplicación móvil](#) del Diseño, se ha implementado la navegación entre interfaces de la aplicación móvil. En la ilustración 6.13 se puede observar la navegación entre las interfaces que componen la aplicación.

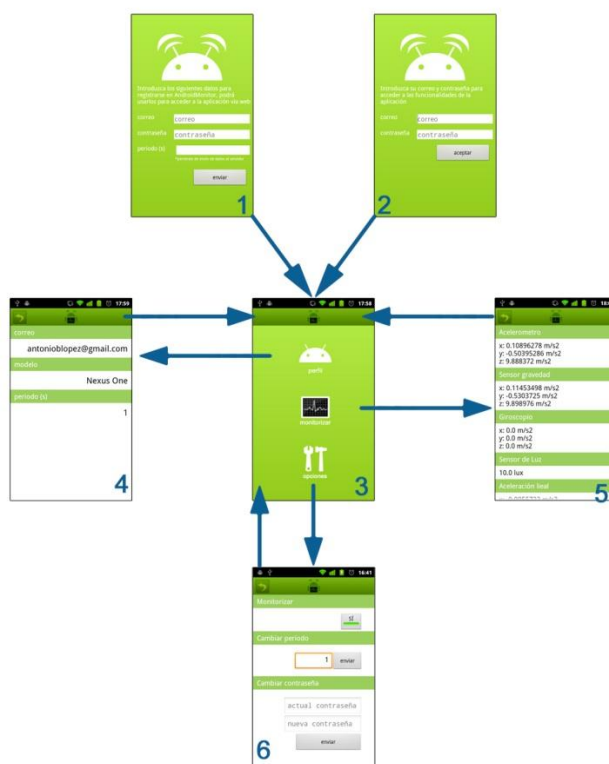


ILUSTRACIÓN 6.13: NAVEGACIÓN DE PANTALLAS EN LA APLICACIÓN MÓVIL

6.2.2.7 Clases auxiliares

A parte de las clases mencionadas en los apartados anteriores, la aplicación móvil cuenta con tres clases auxiliares que contienen funciones para ayudar a realizar las funciones anteriormente descritas. Las clases son `DataBase.java`, `Funciones.java` y `iniciador.java`, clase ya explicada en el [punto 5.3.4 Monitorización](#), estas clase se sitúan en el directorio `/src/es/uc3m/utils`. A continuación se explicará el contenido y finalidad de las dos primeras clases.

6.2.2.7.1 DataBase.java

La clase `DataBase.java` extiende de la clase `SQLiteOpenHelper`, contiene y ejecuta el código que crea la base de datos al instalar la aplicación. Las funciones de la aplicación utilizan esta clase como establecer la conexión con la base de datos y manejar la información que se almacena en ella.

6.2.2.7.2 Funciones.java

La clase `Funciones` incluye una serie de métodos que pueden ser utilizados por las diferentes funcionalidades de la aplicación. Estos métodos son:

```
public String saltRandom() :
```

Esta función genera y devuelve una cadena aleatoria de 64 bytes que se utiliza para la generación de la función resumen de las contraseñas.

```
public String hash(String pass, String salt) :
```

Crea y devuelve la función hash Sha1 de la suma de las cadenas pasadas como parámetros.

```
public String convertStreamToString(InputStream is) :
```

Convierte el objeto de tipo `InputStream` que se le pasa como parámetro a una cadena de texto. Esta función es utilizada para pasar a cadena de texto los mensajes que se recibes desde el servidor como respuestas.

```
public boolean conexionInternet(Context context) :
```

Esta función devuelve un valor booleano que indica si el dispositivo tiene conexión a internet o no. Para ejecutar esta función es necesario que la aplicación cuente con el permiso `android.permission.ACCESS_NETWORK_STATE` declarado en el archivo `Manifest`.

7 Pruebas de evaluación

En este apartado se realizan las pruebas oportunas para verificar el correcto funcionamiento de la aplicación desarrollada.

Para las pruebas se utilizará un servidor y un terminal móvil, en las tablas 5.1 y 5.2 se pueden apreciar las especificaciones de estos sistemas.

TABLA 7.1: ESPECIFICACIONES DEL TERMINAL MÓVIL DE PRUEBA

Modelo	Samsung Galaxy Nexus
CPU	procesador de doble núcleo de 1,2 GHZ
Memoria	1 GB de RAM
Conectividad	HSPA

TABLA 7.2: ESPECIFICACIONES DEL SERVIDOR DE PRUEBA

Procesador	AMD Athlon 64 X2 3800+ / 2.0 GHz
Memoria	1 GB de RAM
Hyper threading	No
Disco Duro	160.0 GB

7.1 Tiempo de ejecución

Lo primero que se realizará será una toma de tiempos para calcular el tiempo de ejecución de los códigos que manejan la monitorización de los dispositivos móviles. Se comenzará con la toma de tiempos del servicio de la aplicación móvil y después se tomará el tiempo de las funciones alojadas en el servidor.

7.1.1 Android Service

La primera prueba que se realizará será calcular el tiempo que tarda el hilo AsyncTask del servicio que toma los valores de los sensores y los envía al servidor. Para ello se utilizará la función `System.currentTimeMillis()` de Java, función que devuelve la hora actual del sistema en milisegundos. Se hará una toma de tiempos en la primera línea del AsyncTask y se hará otra medida al finalizar. A la segunda medida de tiempo se le restará la primera toma y el resultado será el tiempo que ha tardado en milisegundos en ejecutarse el hilo. Se realizarán cinco tomas de tiempo.

ILUSTRACIÓN 7.1: TIEMPO DE EJECUCIÓN DE LA MONITORIZACIÓN DE SENSORES

Número de pruebas	10
Mínimo	0.618 s
Máximo	2.360 s
Media	1.0303 s
Varianza	0,5544 s

Como se puede observar en la tabla anterior el terminal móvil tarda de media 1.0303 segundos en ejecutar el hilo que crea y envía el mensaje de la monitorización al servidor. Que la media sea superior a 1 segundo significa que para monitorizaciones de periodo 1 segundo se podrán perder bastantes mensajes ya que supera el tiempo del periodo. En la prueba también se observa un valor superior a 2 segundos, son valores atípicos pero que suponen pérdida de información que se debería solventar en el futuro.

La siguiente prueba que se realizará será calcular cada cuando se ejecuta la función del servicio, Con esta prueba se comprobará si el código cumple con los periodos introducidos por el usuario. Se realizarán tres pruebas, cada una con una con un periodo distinto con valores de 1 segundo, 2 segundos y 3 segundos. Al igual que en la prueba anterior se realizarán cinco tomas de tiempo. Para realizar la prueba se realizará una toma de tiempo al principio de la ejecución de la función, se guardará esta toma de tiempo en una variable local, en la siguiente iteración se realizará de nuevo la toma y se comprobará la diferencia de tiempo entre la actual toma de tiempo y la almacenada en la variable local, tras esto se actualizará la variable global por la toma actual para la siguiente prueba. En las siguientes tablas se puede comprobar los resultados de estas pruebas.

TABLA 7.3: PRUEBA DE PERIODO PARA 1 SEGUNDO

Número de pruebas	10
Mínimo	0.999 s
Máximo	1.1 s
Media	1 s
Varianza	0,003 s

TABLA 7.4: PRUEBA DE PERIODO PARA 2 SEGUNDOS

Número de pruebas	10
Mínimo	1.999 s
Máximo	2.1 s
Media	2 s
Varianza	0,003 s

TABLA 7.5: PRUEBA DE PERIODO PARA 3 SEGUNDOS

Número de pruebas	10
Mínimo	2.999 s
Máximo	3.1 s
Media	3 s
Varianza	0,003 s

Como se puede apreciar en los resultados la planificación se ha realizado correctamente ya que el tiempo entre ejecución y ejecución difiere del periodo introducido en unas milésimas de segundo.

7.1.2 Servicio Web

En este apartado se calculará el tiempo de ejecución de del código del archivo `moniSens.php` del servidor. Este código es el encargado de recibir los datos que envía el servicio Android y de almacenar estos datos en la base de datos. Para esta prueba se hará uso de la función PHP `microtime(true)` que devuelve la fecha Unix actual en microsegundos. Se realizará una toma de tiempos al inicio de la función y una segunda al finalizar, la diferencia entre la segunda y la primera toma dará como resultado el tiempo de ejecución de la función. Al igual que en las anteriores pruebas se realizarán 5 tomas de tiempo.

TABLA 7.6: TIEMPO DE EJECUCIÓN DEL SERVICIO WEB

Número de pruebas	10
Mínimo	0.467 s
Máximo	0.815 s
Media	0.610 s
Varianza	0,019 s

Como se puede observar los tiempos oscilan entre 0.4 y 0.8 segundos dando como media 0.610 segundos de tiempo de ejecución.

7.1.3 Comunicación

En este punto se realizarán las pruebas de comunicación entre la aplicación Web y la aplicación móvil. El terminal móvil estará conectado a Internet mediante una conexión HSPA.

7.1.3.1 Pérdida de mensajes

En un sistema de monitorización en el que se están enviando mensajes de un sistema a otro constantemente, es normal que alguno de estos mensajes se pierda por varios motivos difícilmente controlables. En este punto se realizará una prueba para calcular el porcentaje de mensajes perdidos. Para la realización de esta prueba se hará dos monitorizaciones de periodo de un segundo durante 30 segundos. El resultado de estas pruebas se puede apreciar en la imagen 7.1

16	0.113	0.010	0.838	2012-09-02 17:43:00	16	0.221	0.069	0.354	2012-09-02 17:59:30
16	0.113	0.009	0.838	2012-09-02 17:43:01	16	0.221	0.068	0.354	2012-09-02 17:59:31
16	0.114	0.008	0.838	2012-09-02 17:43:02	16	0.222	0.068	0.353	2012-09-02 17:59:32
16	0.114	0.009	0.838	2012-09-02 17:43:03	16	0.223	0.068	0.352	2012-09-02 17:59:33
16	0.114	0.007	0.838	2012-09-02 17:43:04	16	0.222	0.068	0.351	2012-09-02 17:59:34
16	0.114	0.008	0.838	2012-09-02 17:43:05	16	0.222	0.068	0.350	2012-09-02 17:59:35
16	0.112	0.010	0.838	2012-09-02 17:43:06	16	0.222	0.068	0.349	2012-09-02 17:59:36
16	0.113	0.009	0.838	2012-09-02 17:43:07	16	0.221	0.066	0.349	2012-09-02 17:59:37
16	0.113	0.008	0.838	2012-09-02 17:43:08	16	0.222	0.067	0.349	2012-09-02 17:59:38
16	0.114	0.007	0.838	2012-09-02 17:43:09	16	0.222	0.067	0.348	2012-09-02 17:59:39
16	0.113	0.008	0.838	2012-09-02 17:43:10	16	0.222	0.068	0.348	2012-09-02 17:59:40
16	0.113	0.007	0.838	2012-09-02 17:43:11	16	0.221	0.067	0.349	2012-09-02 17:59:41
16	0.112	0.004	0.838	2012-09-02 17:43:12	16	0.221	0.067	0.349	2012-09-02 17:59:42
16	0.112	0.002	0.838	2012-09-02 17:43:13	16	0.222	0.067	0.349	2012-09-02 17:59:43
16	0.112	0.003	0.839	2012-09-02 17:43:14	16	0.222	0.067	0.349	2012-09-02 17:59:44
16	0.113	0.003	0.839	2012-09-02 17:43:15	16	0.222	0.067	0.349	2012-09-02 17:59:45
16	0.112	0.002	0.839	2012-09-02 17:43:16	16	0.222	0.068	0.348	2012-09-02 17:59:46
16	0.113	0.005	0.838	2012-09-02 17:43:17	16	0.222	0.069	0.347	2012-09-02 17:59:47
16	0.114	0.005	0.838	2012-09-02 17:43:18	16	0.221	0.069	0.348	2012-09-02 17:59:48
16	0.115	0.004	0.838	2012-09-02 17:43:19	16	0.222	0.069	0.347	2012-09-02 17:59:49
16	0.115	0.004	0.838	2012-09-02 17:43:20	16	0.223	0.069	0.347	2012-09-02 17:59:50
16	0.114	0.003	0.838	2012-09-02 17:43:21	16	0.221	0.068	0.347	2012-09-02 17:59:51
16	0.115	0.003	0.838	2012-09-02 17:43:22	16	0.223	0.068	0.346	2012-09-02 17:59:52
16	0.115	0.002	0.838	2012-09-02 17:43:23	16	0.223	0.068	0.346	2012-09-02 17:59:53
16	0.116	0.001	0.838	2012-09-02 17:43:24	16	0.222	0.068	0.346	2012-09-02 17:59:54
16	0.115	0.001	0.838	2012-09-02 17:43:25	16	0.223	0.068	0.347	2012-09-02 17:59:55
16	0.115	0.001	0.838	2012-09-02 17:43:26	16	0.222	0.069	0.347	2012-09-02 17:59:56
16	0.115	0.002	0.837	2012-09-02 17:43:27	16	0.222	0.068	0.347	2012-09-02 17:59:57
16	0.116	0.002	0.837	2012-09-02 17:43:28	16	0.223	0.068	0.347	2012-09-02 17:59:58
16	0.116	0.000	0.838	2012-09-02 17:43:29	16	0.223	0.069	0.346	2012-09-02 17:59:59
16	0.116	0.001	0.838	2012-09-02 17:43:30	16	0.222	0.068	0.346	2012-09-02 18:00:00

ILUSTRACIÓN 7.2: PRUEBAS DE COMUNICACIÓN

Durante los treinta segundos de cada prueba no se ha perdido ningún mensaje. En la monitorización realizada antes de comenzar la prueba se han capturado pérdidas de mensajes debido a tiempos de cómputo superiores a un segundo en el hilo que envía los datos, punto [7.1.1 Android service](#). De esta prueba se puede concluir que todo mensaje que envíe el terminal móvil será recibido y guardado por el servicio Web, esto significa que la única pérdida de información se da en el terminal móvil.

7.1.4 Conclusiones de los análisis

Tras las pruebas realizadas en este capítulo del documento se puede concluir de los resultados obtenidos, que se ha desarrollado un sistema de monitorización correcto que cumple con las expectativas y objetivos que se le exige.

La mayor preocupación que se extrae de las pruebas realizadas es el tiempo de cómputo del hilo que monitoriza y envía los datos en el terminal móvil. En futuras versiones, se deberá buscar la optimización de su tiempo de cómputo buscando la manera de reducirlo a un tiempo de un segundo en el 100% de los casos para asegurar que en ningún caso se pierdan datos.

8 Conclusiones

En este capítulo se expondrán las conclusiones obtenidas del desarrollo del proyecto realizado a lo largo del presente documento. En segundo lugar se analizarán posibles mejoras que puedan implementarse en el futuro sobre el sistema. Finalmente se redactará un presupuesto detallado del coste que ha supuesto el diseño e implementación del sistema.

8.1 Conclusiones obtenidas

A lo largo de este documento se ha podido comprobar como paso por paso se iba diseñando y construyendo un sistema con la finalidad de cumplir los objetivos impuestos en el punto [1.2 Objetivos](#) de este documento. Tal y como se especifica en dicho punto se han desarrollado dos sistemas comunicados entre sí.

Se ha desarrollado una aplicación móvil que funciona correctamente en terminales con el sistema operativo Android. La aplicación monitoriza y envía los datos a un servicio Web alojado en un servidor externo.

Se ha desarrollado un servicio Web que recopila correctamente los datos que recibe y los almacena para su posterior consulta. Se ha diseñado y maquetado una página Web donde los usuarios de la aplicación pueden consultar los datos que se están enviando desde sus dispositivos móviles.

En conclusión se han alcanzado exitosamente los objetivos que se propusieron al inicio de este Trabajo de Fin de Grado.

8.2 Trabajos futuros

El sistema desarrollado en este proyecto es un sistema con una madurez baja. Se puede considerar la base de un sistema mucho mayor que otorgue en un futuro a sus usuarios una gran cantidad de funcionalidades de tal forma que puedan controlar sus dispositivos remotamente como si los estuvieran controlando en sus manos.

Las mejoras más destacables que se pueden implementar al sistema son las siguientes.

Incrementar el número de sistemas monitorizados de los terminales.

A parte de los sensores, los dispositivos móviles cuentan con otros sistemas que puede resultar interesante controlar remotamente. Prácticamente todos los dispositivos cuentan con un GPS, BlueTooth. NFC. Son muchas las aplicaciones que hacen uso del GPS de los dispositivos para localizarlos. Cada año salen al mercado nuevos dispositivos

móviles con nuevas tecnologías y nuevos sensores. Para conocer con más detalle la situación del terminal se podría analizar el estado de la pantalla, si está encendida o activada, a que tipo de red está conectado: Wi-Fi, 3G, HSPA, LTE.

Mandar instrucciones desde el servidor al dispositivo.

Uno de los principios de un sistema de monitorización es poder controlar remotamente algunos de los elementos del sistema monitorizado. Como propuesta más atractiva para la mejora de esta aplicación se incluye el dar la opción a los usuarios de controlar sus dispositivos desde el servidor, mandando instrucciones al terminal para controlarlo.

Idiomas en la aplicación.

La actual versión del sistema sólo cuenta con la opción de múltiples idiomas en la página web. Como mejora futura figuraría añadir esta opción también en la aplicación móvil.

Evaluación exhaustiva de carga

Para una mejor evaluación del sistema es necesario realizar pruebas con múltiples dispositivos enviando datos para simular una situación más realista del funcionamiento de la aplicación. Con esta prueba se calcularía el número máximo de conexiones que permite el sistema desarrollado.

Sistema de reglas para la monitorización de los sensores.

En futuras versiones se puede dar la opción a los usuarios de establecer un sistema de reglas para la monitorización de sensores. En determinadas situaciones, en lugar de establecer que se envíen datos cada cierto periodo, se puede establecer que los datos sean enviados cuando se cumpla cierta condición que se quiera tener controlada, como por ejemplo el cambio en un determinado sensor.

8.3 Presupuesto del proyecto

En esta sección se presenta el coste detallado del proyecto desarrollado.

1.- Autor:

Antonio Becerra López

2.- Departamento:

Informática

3.- Descripción del Proyecto:

-Título	Diseño de un sistema de monitorización de dispositivos móviles
- Duración (meses)	3
Tasa de costes Indirectos	20%

4.- Presupuesto total del Proyecto (valor en Euros):

6.934 Euros

5.- Desglose presupuestario (costes directos):

PERSONAL

Apellidos y nombre	N.I.F.	Categoría	Dedicación (hombres mes) ^{a)}	Coste hombre mes [23]	Coste (Euro)
Becerra López, Antonio		Analista	0,76	3.289,54	2.500,05
Becerra López, Antonio		Diseñador	0,76	2.289,54	1.740,05
Becerra López, Antonio		Programador	0,76	1.694,39	1.287,74
Hombres mes			2,28	Total	5.527,84

1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
Ordenador Intel(R) Core™ i7	800,00	100	3	60	40,00
Servidor AMD Athlon 64	300,00	100	3	60	15,00
Terminal móvil Android	250,00	100	3	60	12,50
Total					67,50

^{d)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
Total		0,00

OTROS COSTES DIRECTOS DEL PROYECTO ^{e)}

Descripción	Empresa	Coste imputable
Conexión a internet	Vodafone	100,00
Luz	Endesa	80,00
Total		180,00

e) Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes:

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	5.528
Amortización	68
Subcontratación de tareas	0
Costes de funcionamiento	180
Costes Indirectos	1.159
Total	6.934

8.4 Planificación

En este punto se incluye la planificación del proyecto.

El proyecto tiene una duración de tres meses. Se han dividido las tareas en Análisis, Diseño, Desarrollo y Testeo Final. Cada una de estas tareas corresponde con los capítulos de este documento.

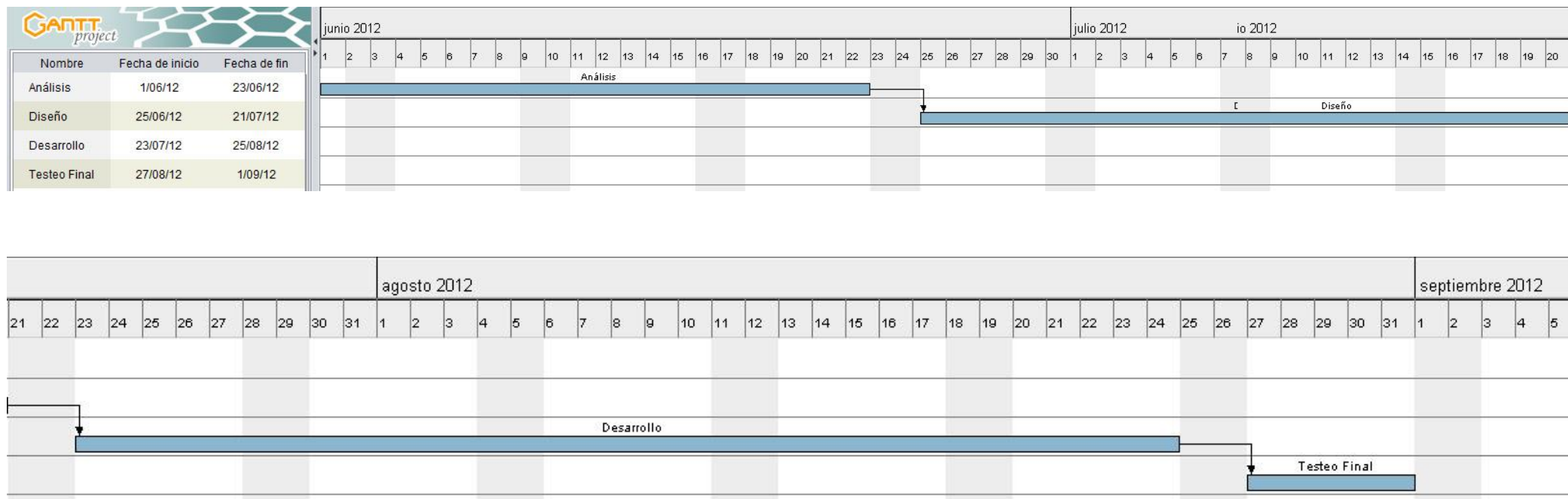


ILUSTRACIÓN 8.1: PLANIFICACIÓN GANTT

9 Bibliografía

- [1] C. Nogueira, «El País,» 26 septiembre 2011. [En línea]. Available: http://elpais.com/diario/2011/09/26/sociedad/1316988001_850215.html. [Último acceso: 2012 agosto 5].
- [2] C. Pettey, «Gartner,» 04 agosto 2011. [En línea]. Available: <http://www.gartner.com/it/page.jsp?id=1759714>. [Último acceso: 05 agosto 2012].
- [3] [x]cube labs, «xcubelabs,» 2011. [En línea]. Available: <http://www.xcubelabs.com/evolution-of-mobile-operating-systems.php>. [Último acceso: 15 marzo 2012].
- [4] [x]cube labs, «xcubelabs,» 2011. [En línea]. Available: <http://www.xcubelabs.com/the-android-story.php>. [Último acceso: 15 marzo 2012].
- [5] Google, «android.com,» 2012. [En línea]. Available: <http://www.android.com/about/jelly-bean/>. [Último acceso: 01 agosto 2012].
- [6] Developer Android, «Developer Android,» 2012. [En línea]. Available: <http://developer.android.com/resources/dashboard/platform-versions.html>. [Último acceso: agosto 2012].
- [7] Developer Android, «Developer Android,» 2012. [En línea]. Available: <http://developer.android.com/guide/basics/what-is-android.html>. [Último acceso: 16 marzo 2012].
- [8] Android Developer, «developer.android.com,» 14 agosto 2012. [En línea]. Available: <http://developer.android.com/intl/es/reference/android/app/Activity.html>.

[Último acceso: 20 agosto 2012].

- [9] Apple, «apple.com,» 2012. [En línea]. Available: <http://www.apple.com/es/ios/ios6/>. [Último acceso: 01 agosto 2012].
- [10] Apple, «apple.com/security,» 2012. [En línea]. Available: <http://www.apple.com/la/business/accelerator/develop/security.html>. [Último acceso: 01 agosto 2012].
- [11] Microsoft, «Microsoft.com,» 22 marzo 2012. [En línea]. Available: <http://msdn.microsoft.com/en-us/library/ff402531%28v=vs.92%29.aspx>. [Último acceso: 06 agosto 2012].
- [12] F. Ruiz, «Culturacion,» 18 mayo 2012. [En línea]. Available: <http://culturacion.com/2012/05/blackberry-sistema-operativo-movil-de-rim/>. [Último acceso: 06 agosto 2012].
- [13] Ecured, «Ecured,» 2012. [En línea]. Available: http://www.ecured.cu/index.php/Symbian_OS. [Último acceso: 6 agosto 2012].
- [14] Nokia, «Developer Nokia,» 2009. [En línea]. Available: http://library.developer.nokia.com/index.jsp?topic=/S60_3rd_Edition_Cpp_Developers_Library/GUID-4BB43883-5767-4A50-BD53-68145D16D9D8.html. [Último acceso: 05 agosto 2012].
- [15] HP WebOS, «Developer Palm,» 2011. [En línea]. Available: https://developer.palm.com/content/resources/develop/overview_of_webos/overview_of_webos_palm_webos_architecture.html. [Último acceso: 05 agosto 2012].
- [16] Consultoriajava, «consultoriajava,» [En línea]. Available: <http://www.consultoriajava.com/tools/axis.shtml>. [Último acceso: 7 agosto 2012].
- [17] J. Louvel, T. Templier y . T. Boil, Restlet in Action: Developing RESTful web APIs in

Java, Manning, 2012.

- [18] R. Navarro Marset, «REST vs Web Services,» 17 septiembre 2007. [En línea]. Available:
<http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>.
[Último acceso: 13 agosto 2012].
- [19] Automatas.org, «automatas.org,» 02 marzo 2006. [En línea]. Available:
<http://automatas.org/redes/scadas.htm>. [Último acceso: 06 agosto 2016].
- [20] H. Mendiburu Diaz, «Galeon,» 30 septiembre 2005. [En línea]. Available:
<http://www.galeon.com/hamd/pdf/scada.pdf>. [Último acceso: 6 agosto 2012].
- [21] Agencia Española de Protección de Datos, «agpd.es,» 2010. [En línea]. Available:
http://www.agpd.es/portalwebAGPD/jornadas/dia_proteccion_2011/responsable/index-ides-idphp.php. [Último acceso: 15 agosto 2012].
- [22] K. Beck, Extreme Programming Explained: Embrace Change, US ed, 1999.
- [23] tufuncion, «tufuncion,» 7 abril 2008. [En línea]. Available:
<http://www.tufuncion.com/trabajo-programador>. [Último acceso: 20 agosto 2012].

10 Anexos

En este capítulo se expondrán los anexos de la memoria del presente Proyecto de Fin de Grado. En el primer anexo se expondrán los manuales de instalación. En el segundo anexo se expondrán los manuales de usuario de las aplicaciones.

10.1 Anexo I: Manual de instalación

En este anexo se explicará paso a paso la forma de instalar el sistema.

10.1.1 Aplicación web

En esta sección se explicará como crear un servidor y los preparativos necesarios que se han de realizar para alojar en él la web del proyecto.

10.1.1.1 Instalación del servidor

El servidor se montará sobre el sistema operativo Ubuntu Server en su versión 12.04, sistema que se puede descargar gratuitamente desde la página web ubuntu.com. Una vez instalado el sistema operativo se comenzará a preparar el sistema para alojar una página Web. Dado que la mayoría de las instrucciones que se deberán escribir en la consola requerirán permisos de administración lo primero que se realizará será cambiar en la consola el usuario al usuario root con el comando:

```
sudo su
```

Este comando solicitará la contraseña que se introdujo al instalar el sistema operativo. Lo primero que se realizará será cambiar el modo de asignación de IP para el equipo, para esto será necesario modificar el archivo `interfaces`, pero antes de modificarlo es conveniente crear una copia de seguridad con la siguiente línea de código.

```
cp /etc/network/interfaces /etc/network/interfacesOLD
```

Con esta línea se ha copiado el archivo `interfaces` al archivo `interfacesOLD`. A continuación se modificará el archivo `interfaces`.

```
nano /etc/network/interfaces
```

Esta línea de código mostrará en pantalla el contenido del archivo `interfaces`, ilustración 10.1, que se debe modificar.

```

GNU nano 2.2.6      File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

[ Read 10 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell

```

ILUSTRACIÓN 10.1: MANUAL DE INSTALACIÓN: ARCHIVO INTERFACES

Se deberá modificar añadiendo los parámetros que deseemos para nuestro servidor. Lo primero que se deberá cambiar será las siglas dhcp de la última línea por la palabra static. A continuación se especificará cual será la dirección IP fija de la máquina con la línea “address <IP>”, tras esta línea se establecerá la máscara de red con la línea “netmask 255.255.255.0” finalmente la IP de la puerta de enlace con la línea “Gateway <IP>”. El resultado debería ser similar al mostrado en la ilustración 10.2.

```

GNU nano 2.2.6      File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static

# IP fija
address 192.168.19.131
# Mascara de subred
netmask 255.255.255.0
# IP router
gateway 192.168.19.1

[ Read 17 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell

```

ILUSTRACIÓN 10.2: MANUAL DE INSTALACIÓN: ARCHIVO INTERFACES MODIFICADO

Al terminar de editar se deberá pulsar F2, la tecla ‘y’ para guardar y ‘enter’ para confirmar. Finalmente se reinicia la red para efectuar los cambios.

/etc/init.d/networking restart

Para el servidor se utilizará LAMP, un conjunto de paquetes que equiparan a nuestro servidor del servidor web Apache2, servidor de bases de datos MySQL 5.1 y soporte para PHP5, para instalarlo se hará uso de la siguiente línea.

```
sudo taskset install lamp-server
```

El sistema comenzará a descargar los paquetes hasta que solicite la contraseña del usuario root para MySQL, ilustración 10.3.

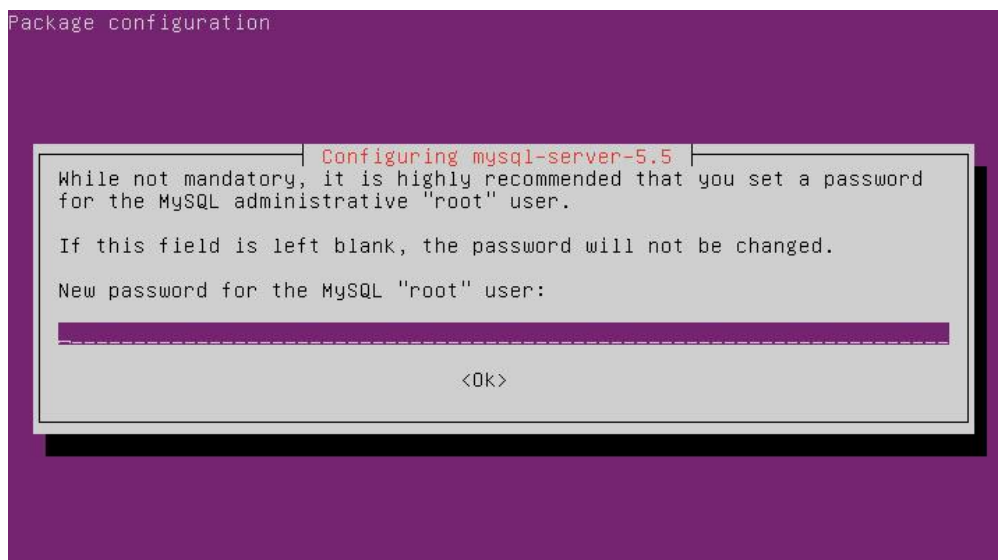


ILUSTRACIÓN 10.3: MANUAL DE INSTALACIÓN: LAMP

Para comprobar que se ha instalado correctamente se escribirá en un navegador web la IP de la máquina lo que hará que se muestre el texto "It works!" en el navegador, ilustración 10.4.

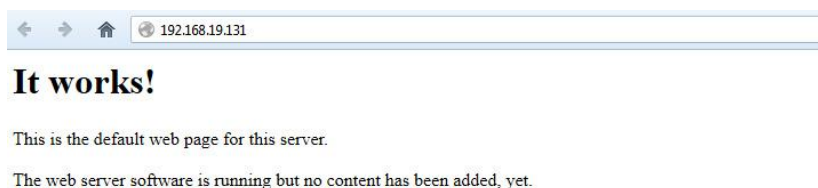


ILUSTRACIÓN 10.4: MANUAL DE INSTALACIÓN: SERVIDOR WEB

A continuación se procederá a instalar el administrador de bases de datos MySQL llamado PhpMyAdmin, para ello se escribirá la siguiente línea en la consola de comandos.

```
sudo aptitude install phpmyadmin
```

El sistema comenzará la instalación del programa hasta que llegue un punto donde solicite elegir para qué tipo de servidor Web se desea que se adapte la instalación. Se deberá elegir Apache2, ilustración 40.5

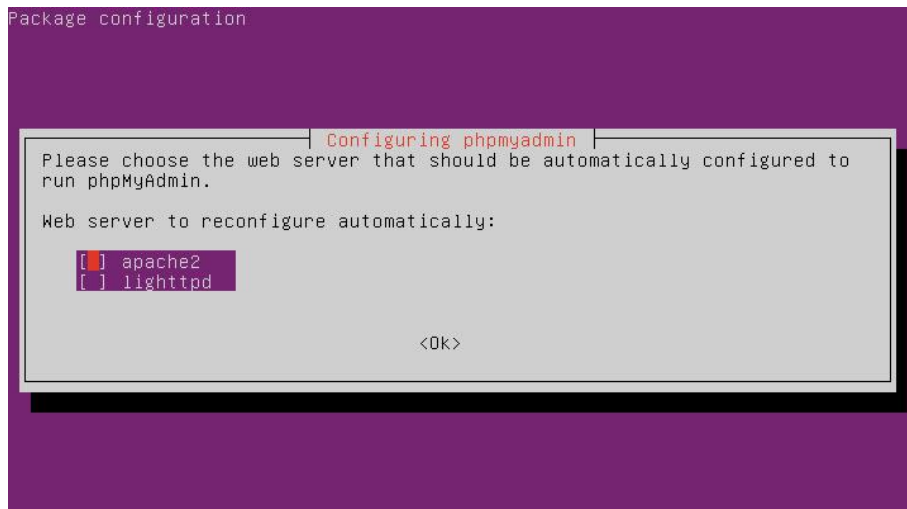


ILUSTRACIÓN 10.5: MANUAL DE INSTALACIÓN: PHPMYADMIN INSTALACIÓN

Al continuar con la instalación más delante, el sistema solicitará las contraseñas de acceso para PhpMyAdmin, ilustración 10.6.

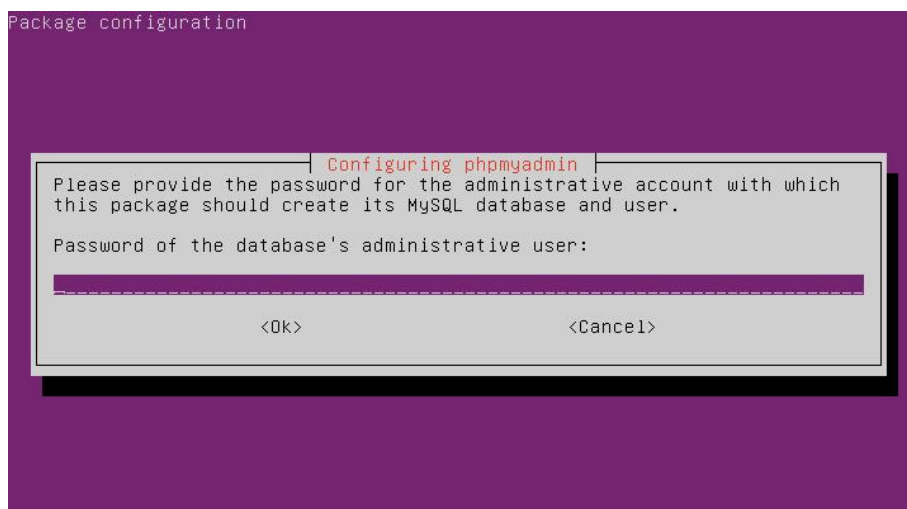


ILUSTRACIÓN 10.6: MANUAL DE INSTALACIÓN: PHPMYADMIN INSTALACIÓN 2

10.1.1.2 Importación de la base de datos

Para subir la base de datos al servidor se importará el código en PhpMyAdmin. Se deberá acceder a la página del administrador de PhpMyAdmin en el servidor situado en la dirección `http://<miIP>/phpmyadmin` al acceder aparecerá un login, ilustración 10.7. Para acceder se deberán introducir los datos que se seleccionaron al realizar la instalación y pulsar “Continuar”.



ILUSTRACIÓN 10.7: MANUAL DE INSTALACIÓN: PHPMYADMIN LOGIN

Una vez se haya accedido a la aplicación se ha de crear la base de datos “androidmonitor” para ello se ha de pulsar la pestaña “Bases de datos” y crearla desde ahí, ilustración 10.8.



ILUSTRACIÓN 10.8: MANUAL DE INSTALACIÓN: CREAR LA BASE DE DATOS

Una vez creada se ha de seleccionar la base creada de la lista de bases de datos a la izquierda de la pantalla, tras seleccionarla se ha de pulsar sobre la pestaña “Importar”. En la pestaña abierta, ilustración 10.9, se ha de seleccionar el código de la base de datos incluido en el disco adjunto a este documento, el código se encuentra en el archivo `androidmonitor.sql` en el directorio `/src/bbdd/` del disco adjunto. Finalmente se ha de pulsar el botón Continuar para que se realice la importación.

ILUSTRACIÓN 10.9: MANUAL DE INSTALACIÓN: IMPORTAR CÓDIGO

10.1.1.3 Carga de la web

Para cargar la web en el servidor Web se utilizará el Protocolo de Transferencia de Archivos o FTP. Antes de cargar la web se deberá modificar los datos de la conexión con la base de datos que se encuentra en el archivo `/src/web/php_code/conexión.php` del disco que se incluye con este documento. En este documento se ha de modificar los datos de la función `reset` por los propios. Una vez cambiado estos datos se procederá a subir los datos al servidor mediante un cliente FTP como FileZilla, software gratuito que se puede descargar desde filezilla-project.org. Una vez abierto FileZilla se deberá crear una nueva conexión con los datos de conexión a la FTP establecidos al crear el servidor y pulsar sobre el botón conectar. Una vez conectado al servidor se deberán subir todos los archivos situados en el directorio `/src/web/`, del disco incluido en la entrega, al directorio principal del servidor, `/www/` o a la carpeta deseada dentro de este directorio, ilustración 10.10.

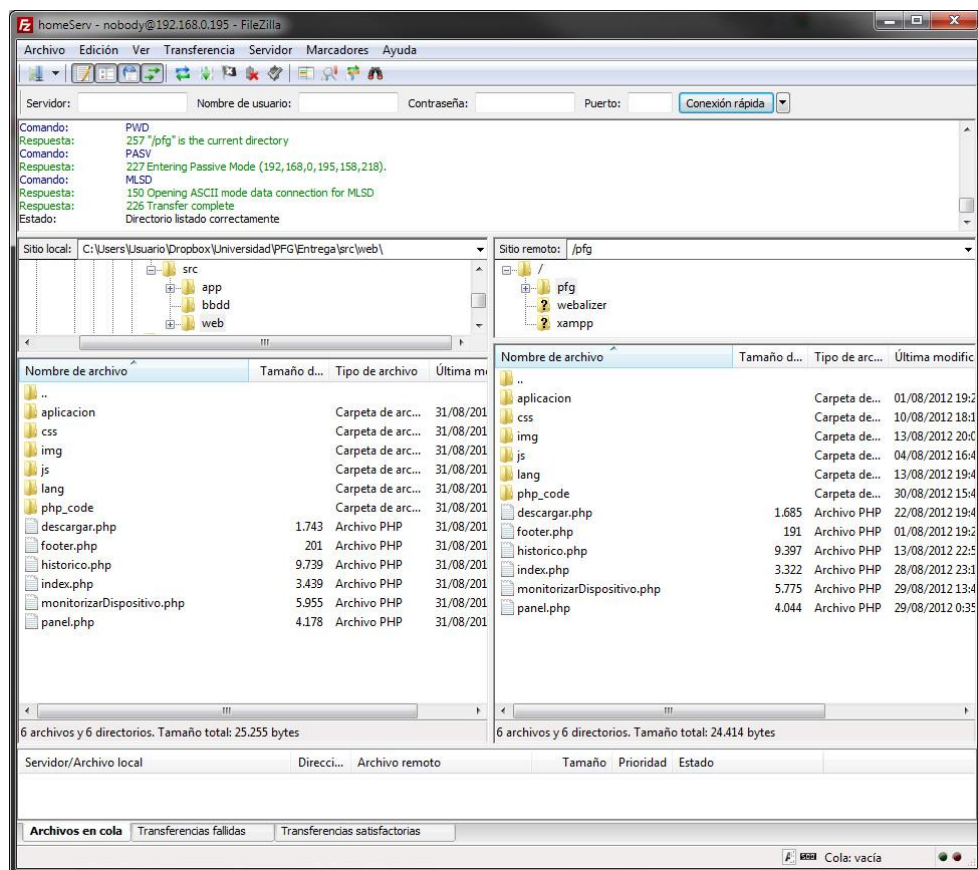


ILUSTRACIÓN 10.10: MANUAL DE INSTALACIÓN: SUBIDA DE LA WEB

10.1.2 Aplicación móvil

En este apartado se procederá a explicar como instalar y preparar el entorno de desarrollo para la aplicación móvil.

10.1.2.1 Preparación del entorno de desarrollo

Para el desarrollo de esta aplicación se utilizará el entorno de desarrollo integrado de código abierto y multiplataforma Eclipse, software desarrollado por Eclipse Foundation, y el Android SDK proporcionado por la compañía Google.

Eclipse se descargará desde la página oficial del proyecto eclipse.org, se ha de descargar la versión para desarrolladores de Java y para su sistema operativo correspondiente, Windows, Mac OS o Linux.

Eclipse no requiere instalación, por lo que una vez descargado el archivo *zip* tan solo se deberá extraer su contenido en un directorio.

Lo siguiente que se deberá hacer será descargar el SDK de Android correspondiente a su sistema operativo de la página developer.android.com, una vez descargado se deberá descomprimir en un directorio, se deberá ejecutar la aplicación SDK Manager e instalar los paquetes y actualizaciones.

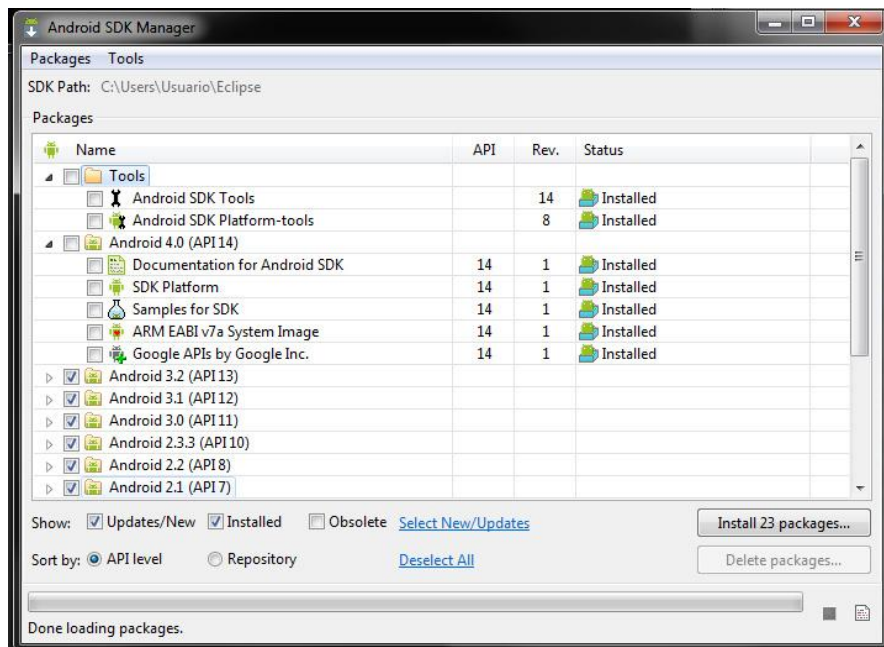


ILUSTRACIÓN 10.11: MANUAL DE INSTALACIÓN: ANDROID SDK MANAGER

El siguiente paso a realizar será integrar el SDK de Android con Eclipse, para ello se tendrán que seguir los siguientes pasos.

Acceder en el menú superior de eclipse a Help > Install New Software...

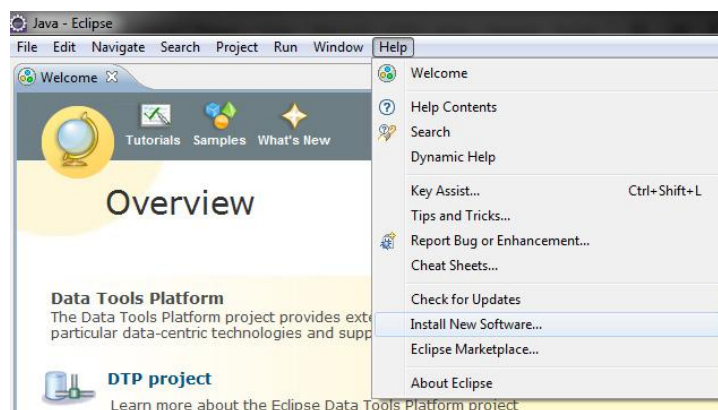


ILUSTRACIÓN 10.12: MANUAL DE INSTALACIÓN: INSTALAR NUEVO SOFTWARE EN ECLIPSE

Pulsar el botón Add... situado en la parte superior derecha de la ventana que se abrirá lo que hará que se abra una nueva ventana, escribir en la casilla Name el nombre del plugin a instalar, ADT plugin, y en Location la siguiente url 'https://dl-ssl.google.com/android/eclipse/' después pulsar OK.

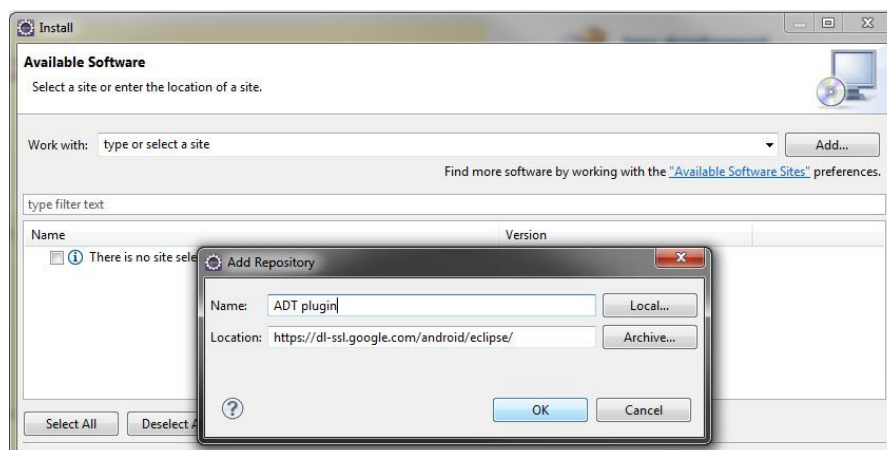


ILUSTRACIÓN 10.13: MANUAL DE INSTALACIÓN: INSTALAR EL PLUGIN

Seleccionar los paquetes encontrados y comenzar su instalación pulsando el botón Next

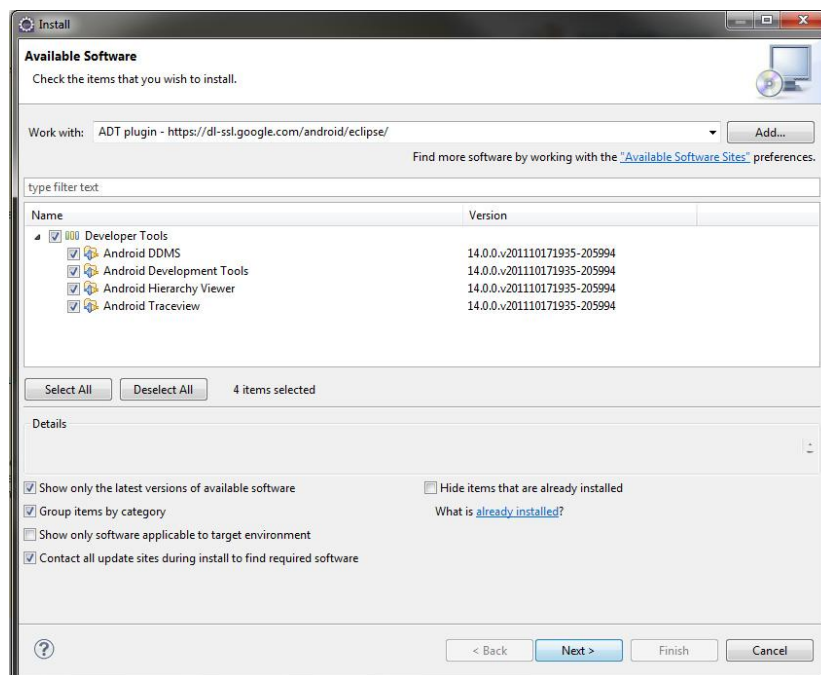


ILUSTRACIÓN 10.14: MANUAL DE INSTALACIÓN: SELECCIÓN DE PAQUETES

Una vez finalizada la instalación se requerirá el reinicio de Eclipse



ILUSTRACIÓN 10.15: MANUAL DE INSTALACIÓN: CONFIGURAR SDK

10.1.2.2 Importación de la aplicación

Una vez instalado el SDK de Android en Eclipse se podrán crear nuevos proyectos Android desde el menú superior File > New >Project... AndroidMonitor no es un proyecto que deba ser creado desde 0, el disco que se adjunta con este documento

incluye su código fuente por lo que se deberá importar este código para empezar a trabajar. En la ventana que se abra al pulsar sobre “Proyect...”, ilustración 10.16, se deberá abrir la carpeta “Android” y seleccionar “Android Project from Existing Code”.

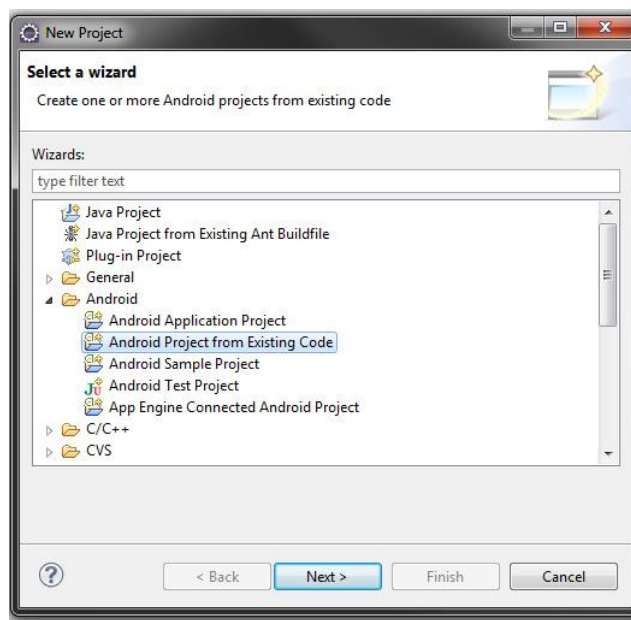


ILUSTRACIÓN 10.16: MANUAL DE INSTALACIÓN: IMPORTAR PROYECTO 1

Al pulsar sobre el botón “Next” se abrirá una nueva ventana, ilustración 10.17. Se deberá pulsar el botón “browser” para seleccionar la carpeta app situada en el directorio /src del disco adjunto a este documento. Una vez seleccionado se deberá pulsar sobre la casilla “Copy project into workspace” para activarla. Tras esto se pulsará sobre el botón finish.

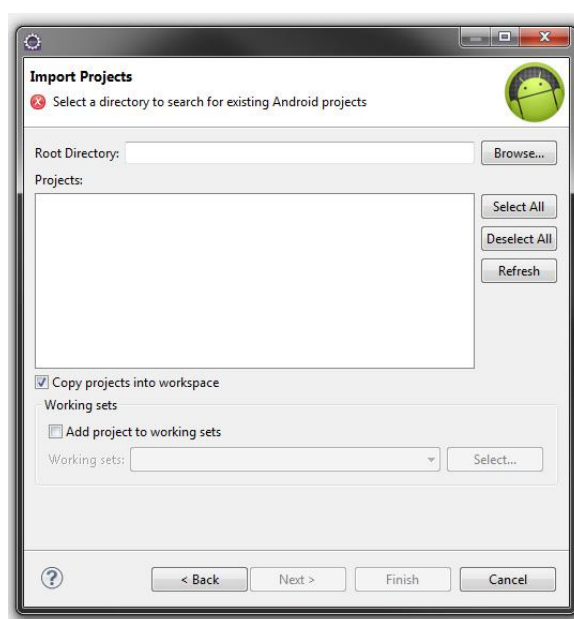


ILUSTRACIÓN 10.17: MANUAL DE INSTALACIÓN: IMPORTAR PROYECTO 2

10.2 Anexo II: Manual de usuario

En este anexo se expondrán los manuales de usuario que servirán como guía a los usuarios de la aplicación para conocer y aprender a ejecutar las diferentes funcionalidades que ofrece el sistema.

10.2.1 Aplicación móvil

En esta sección se expondrá el manual de usuario de la aplicación móvil. En primer lugar se expondrá el funcionamiento y las funcionalidades de la aplicación, a continuación se tratarán de dar solución a los problemas más comunes y finalmente se darán respuesta a las preguntas más frecuentes.

10.2.1.1 Iniciar aplicación

Para registrarte en la aplicación se deberá iniciar la aplicación pulsando sobre el icono de AndroidMonitor en el menú de aplicación del dispositivo Android, ilustración 10.18. Para que el icono aparezca en este menú se ha debido instalar la aplicación en el terminal correctamente. Al pulsar sobre el icono la aplicación se iniciará y se mostrará en pantalla.



ILUSTRACIÓN 10.18: MANUAL DE USUARIO: MENÚ ANDROID 2.3.6

10.2.1.2 Registro en la aplicación

Si es la primera vez que se inicia la aplicación, punto [10.2.1.1 Iniciar aplicación](#), la pantalla que se mostrará a continuación de pulsar el icono será la pantalla de registro, ilustración 10.19, en esta pantalla se mostrará un formulario con tres campos: correo, contraseña y periodo. Para realizar el registro se han de rellenar los tres campos de este formulario y pulsar en el botón enviar situado en la parte inferior derecha de la pantalla.

La ilustración muestra una pantalla de registro con un fondo verde. En la parte superior hay un icono de un robot Android con antenas. Debajo del icono, un texto indica: "Introduzca los siguientes datos para registrarse en AndroidMonitor, podrá usarlos para acceder a la aplicación via web". A continuación, hay tres campos de entrada: "correo" con el texto "correo" dentro, "contraseña" con el texto "contraseña" dentro, y "periodo (s)" que está vacío. Debajo del campo "periodo (s)" hay un texto pequeño: "*periodo de envío de datos al servidor". En la parte inferior derecha hay un botón gris con el texto "enviar".

ILUSTRACIÓN 10.19: MANUAL DE USUARIO: REGISTRO EN LA APLICACIÓN

En el campo correo se ha de introducir una dirección de correo electrónico. En el campo contraseña se ha de introducir la contraseña que se desea tener para acceder a la aplicación. En el campo periodo se debe introducir un valor numérico mayor que 0 que marcará el periodo de la monitorización del sistema. Si se han introducido los datos correctamente al pulsar el botón enviar el usuario será redirigido a la pantalla menú, ilustración 10.22, si se ha introducido algún dato incorrectamente se mostrará un mensaje por pantalla, ilustración 10.20.



ILUSTRACIÓN 10.20: MANUAL DE USUARIO: DATO MAL INTRODUCIDO.

10.2.1.3 Acceso a la aplicación

Si ya se ha realizado anteriormente el registro en la aplicación, punto [10.2.1.2 Registro en la aplicación](#), al iniciar la aplicación, punto [10.2.1.1 Iniciar aplicación](#), se mostrará la pantalla de identificación, ilustración 10.21. En esta pantalla se mostrará un formulario con dos campos: correo y contraseña; Para acceder a la aplicación se deberán rellenar ambos campos con los mismos datos introducidos al realizar el registro.



ILUSTRACIÓN 10.21: MANUAL DE USUARIO: IDENTIFICACIÓN DE USUARIOS

Si se han introducido los datos correctamente al pulsar el botón enviar el usuario será redirigido a la pantalla menú, ilustración 10.22, si se ha introducido algún dato incorrectamente se mostrará un mensaje por pantalla, ilustración 10.20.

10.2.1.4 Ver Perfil

Una vez realizado el acceso a la aplicación correctamente, punto [10.2.1.3 Acceso a la aplicación](#), en la pantalla del terminal se mostrará el menú principal de la aplicación, ilustración 10.22.

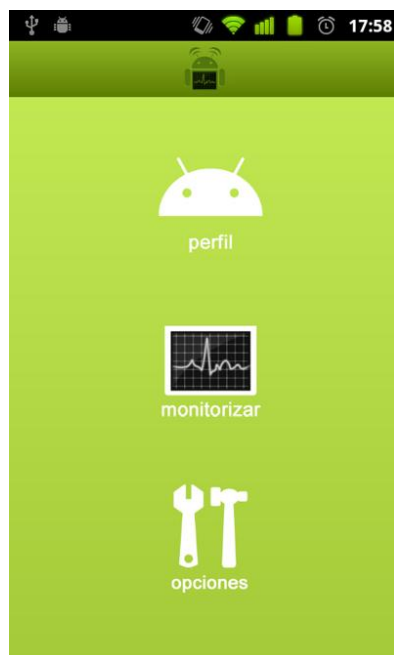


ILUSTRACIÓN 10.22: MANUAL DE USUARIO: MENÚ PRINCIPAL

En este menú se mostrarán tres iconos colocados en fila uno encima de otro, desde arriba son: perfil, monitorizar y opciones. Para acceder al perfil se deberá pulsar sobre el primer icono empezando por arriba. Al pulsar sobre este icono en la pantalla se mostrará la pantalla del perfil, ilustración 10.23.

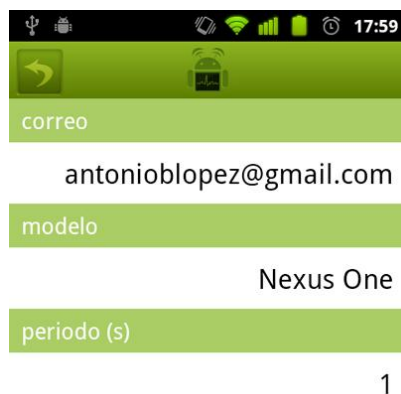


ILUSTRACIÓN 10.23: MANUAL DE USUARIO: PAGINA PERFIL DE USUARIO

En esta página se podrán ver los datos: correo electrónico, modelo del dispositivo y periodo asignado. De estos tres datos el único que puede ser cambiado es el periodo, para cambiarlo se ha de hacer desde la pantalla opciones, punto [10.2.1.6 ver opciones](#). Para volver al menú principal se deberá pulsar el botón con la flecha situado en la esquina superior izquierda de la pantalla.

10.2.1.5 Monitorizar

Para acceder a la monitorización de sensores desde el terminal se ha debido acceder a la aplicación, punto [10.2.1.3 Acceso a la aplicación](#), y encontrarse en el menú principal de la aplicación, ilustración 10.22. En este menú se debe pulsar en el segundo icono, monitorizar, al pulsarlo la pantalla se redireccionará a la pantalla Monitorizar, ilustración 10.24.

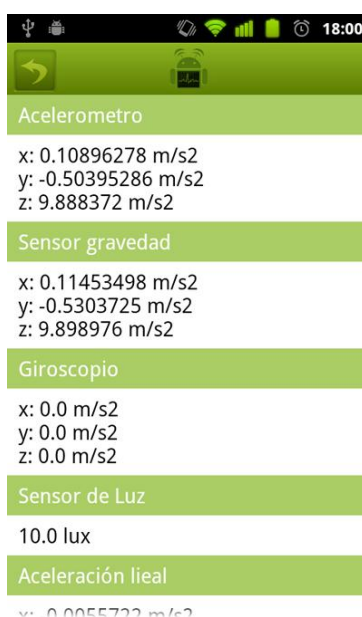


ILUSTRACIÓN 10.24: MANUAL DE USUARIO: PANTALLA MONITORIZAR

Para volver al menú principal se deberá pulsar el botón con la flecha situado en la esquina superior izquierda de la pantalla.

10.2.1.6 Ver opciones

Para acceder a la monitorización de sensores desde el terminal se ha debido acceder a la aplicación, punto [10.2.1.3 Acceso a la aplicación](#), y encontrarse en el menú principal de la aplicación, ilustración 10.22. En el menú se deberá pulsar en el tercer icono, opciones, para ser redirigido a la pantalla de opciones de la aplicación.

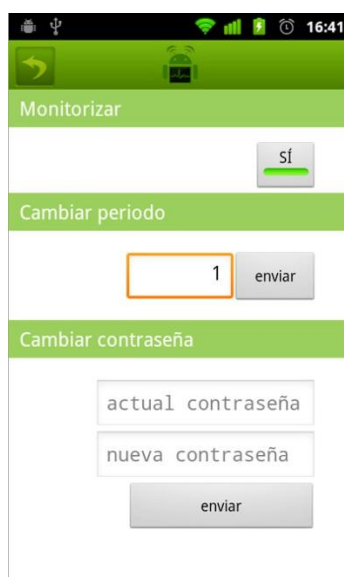


ILUSTRACIÓN 10.25: MANUAL DE USUARIO: PANTALLA OPCIONES

Para volver al menú principal se deberá pulsar el botón con la flecha situado en la esquina superior izquierda de la pantalla.

10.2.1.6.1 Activa o desactivar monitorización

Para activar o desactivar la monitorización se debe haber accedido a la pantalla opciones, punto [10.2.1.6 Ver opciones](#). El estado de la monitorización lo marca el botón situado debajo del título “Monitorizar”, ilustración 10.25. Para cambiar el estado se deberá pulsar sobre el mismo botón.

10.2.1.6.2 Cambiar periodo

Para cambiar el periodo se debe haber accedido a la pantalla opciones, punto [10.2.1.6 Ver opciones](#). El actual valor del periodo se encuentra en el campo de texto situado debajo del título “Cambiar periodo”, ilustración 10.25. Para cambiar este valor se debe pulsar sobre el campo e introducir el nuevo valor que se desee, una vez introducido pulsar sobre el botón enviar situado al lado derecho de este campo.

10.2.1.6.3 Cambiar contraseña

Para cambiar la contraseña se debe haber accedido a la pantalla opciones, punto [10.2.1.6 Ver opciones](#). El cambio de contraseña se realiza mediante el formulario situado debajo del título “Cambiar contraseña”, ilustración 10.25. Este formulario está compuesto de dos campos, en el primer campo se ha de introducir la contraseña actual de la aplicación, en el segundo campo se ha de introducir la nueva contraseña deseada. Una vez rellenados los dos campos se ha de pulsar sobre el botón aceptar situado

debajo de estos campos. Para cambiar la contraseña es necesario que el dispositivo tenga conexión a Internet.

10.2.1.7 Resolución de problemas

En este punto se resolverán los principales problemas que puedan surgir en el uso de la aplicación.

10.2.1.7.1 No puedo acceder a la aplicación.

- Asegúrese de haber introducido correctamente los datos de solicitados, el correo electrónico y la contraseña que utilizó para registrarse en la aplicación.

10.2.1.7.2 No se actualizan los datos en la pantalla de monitorizar

- Asegúrese de tener activada la monitorización en el menú de opciones, punto [10.2.1.6.1 Activa o desactivar monitorización](#).

10.2.1.7.3 No puedo cambiar la contraseña.

- Asegúrese de haber introducido correctamente la contraseña actual, la contraseña que ha usado para acceder a la aplicación.
- Asegúrese que tiene conexión a Internet. Es necesario tener conexión a Internet para poder cambiar la contraseña.
- Es posible que el servidor web no se encuentre disponible en ese momento, intente cambiar más adelante la contraseña.

10.2.1.8 Preguntas más frecuentes

En este punto se darán respuestas a las preguntas más frecuentes que pueden surgir del uso de la aplicación.

10.2.1.8.1 ¿Puede acceder un usuario no registrado a la aplicación?

- No. Para acceder a la aplicación se ha debido realizar el registro con anterioridad.

10.2.2 Aplicación Web

En esta sección se expondrá el manual de usuario de la página web. En primer lugar se expondrá el funcionamiento y las funcionalidades de la aplicación, a continuación se tratarán de dar solución a los problemas más comunes y finalmente se darán respuesta a las preguntas más frecuentes.

10.2.2.1 Acceso a la aplicación

Al acceder a la web lo primero que se mostrará será un formulario con dos campos, correo y contraseña, ilustración 10.26. Este formulario sirve para identificarse y poder acceder a la web. Para acceder a la web se deben introducir en estos campos los datos que se introdujeron al realizar el registro en la aplicación móvil.

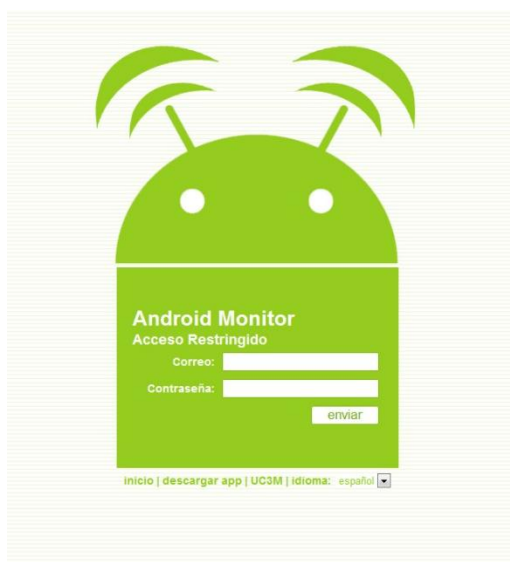
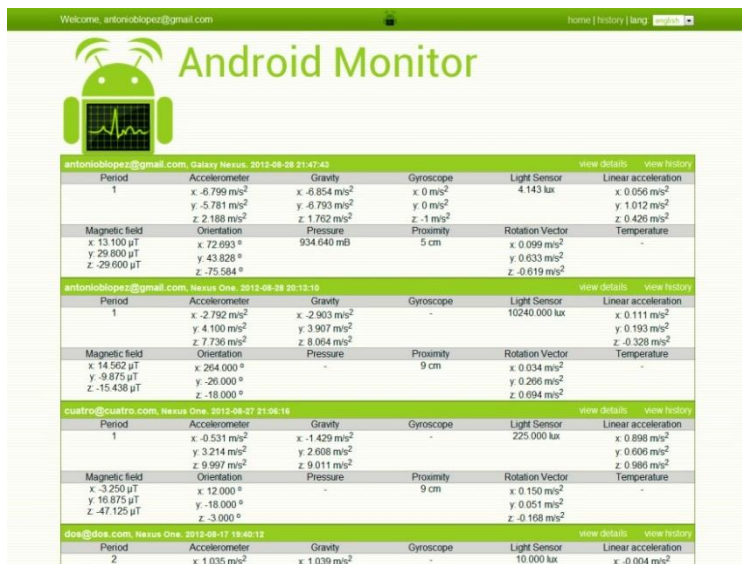


ILUSTRACIÓN 10.26: MANUAL DE USUARIO: IDENTIFICACIÓN EN LA PÁGINA WEB

Una vez introducidos los datos al pulsar sobre el botón enviar el usuario será redirigido a la página panel, ilustración 10.27.




Welcome, antonioblopez@gmail.com									
 <h1>Android Monitor</h1>									
antonioblopez@gmail.com, Nexus One, 2012-08-28 21:47:43									
Period	Accelerometer	Gravity	Gyroscope	Light Sensor	Linear acceleration				
1	x: -6.799 ms ⁻² y: -5.781 ms ⁻² z: 2.188 ms ⁻²	x: -6.854 ms ⁻² y: -6.793 ms ⁻² z: 1.762 ms ⁻²	x: 0 ms ⁻² y: 0 ms ⁻² z: -1 ms ⁻²	4.143 lux	x: 0.056 ms ⁻² y: 0.012 ms ⁻² z: 0.426 ms ⁻²				
Magnetic field	x: 13.100 μT y: 29.800 μT z: -29.600 μT	Orientation	Pressure	Proximity	Rotation Vector	Temperature			
	x: 72.693 ° y: 43.828 ° z: -75.584 °		934.640 mB	5 cm	x: 0.099 ms ⁻² y: 0.633 ms ⁻² z: -0.619 ms ⁻²				
antonioblopez@gmail.com, Nexus One, 2012-08-29 10:15:10									
Period	Accelerometer	Gravity	Gyroscope	Light Sensor	Linear acceleration				
1	x: -2.792 ms ⁻² y: 4.100 ms ⁻² z: 7.736 ms ⁻²	x: -2.903 ms ⁻² y: 3.907 ms ⁻² z: 8.064 ms ⁻²	-	10240.000 lux	x: 0.111 ms ⁻² y: 0.193 ms ⁻² z: -0.328 ms ⁻²				
Magnetic field	x: 14.502 μT y: -9.875 μT z: -15.438 μT	Orientation	Pressure	Proximity	Rotation Vector	Temperature			
	x: 264.000 ° y: -26.000 ° z: -18.000 °		-	9 cm	x: 0.034 ms ⁻² y: 0.286 ms ⁻² z: 0.694 ms ⁻²				
cuatro@cuatro.com, Nexus One, 2012-08-27 21:04:16									
Period	Accelerometer	Gravity	Gyroscope	Light Sensor	Linear acceleration				
1	x: -0.531 ms ⁻² y: 3.214 ms ⁻² z: 9.997 ms ⁻²	x: -1.429 ms ⁻² y: 2.608 ms ⁻² z: 9.011 ms ⁻²	-	225.000 lux	x: 0.898 ms ⁻² y: 0.606 ms ⁻² z: 0.986 ms ⁻²				
Magnetic field	x: -3.250 μT y: 16.875 μT z: -47.125 μT	Orientation	Pressure	Proximity	Rotation Vector	Temperature			
	x: 12.000 ° y: -18.000 ° z: -3.000 °		-	9 cm	x: 0.150 ms ⁻² y: 0.051 ms ⁻² z: -0.168 ms ⁻²				
dos@dos.com, Nexus One, 2012-08-17 19:40:12									
Period	Accelerometer	Gravity	Gyroscope	Light Sensor	Linear acceleration				
2	x: 1.035 ms ⁻² y: -0.004 ms ⁻² z: -0.004 ms ⁻²	x: 1.039 ms ⁻² y: -0.004 ms ⁻² z: -0.004 ms ⁻²	-	10.000 lux	x: -0.004 ms ⁻² y: -0.004 ms ⁻² z: -0.004 ms ⁻²				

ILUSTRACIÓN 10.27: MANUAL DE USUARIO: PÁGINA PANEL DE LA WEB

10.2.2.2 Descargar aplicación móvil.

Si no se ha realizado el registro en la aplicación móvil no se podrá acceder a la página web. Por esto en la página hay una sección desde la cual se puede descargar la aplicación móvil para instalarla y realizar el registro. Para descargarla se deberá acceder a la página descarga, ilustración 10.28, desde el enlace “descarga app” situado debajo del formulario de identificación, ilustración 10.26.



ILUSTRACIÓN 10.28: MANUAL DE USUARIO: DESCARGAR APLICACIÓN

La descarga de la aplicación se podrá realizar de dos formas. Pulsando sobre la imagen del código, al pulsar se iniciará la descarga en el dispositivo desde el cual se está visitando la página. Capturando el código con una aplicación lectora de código, como

por ejemplo Google Goggles, desde el teléfono en el cual se quiere descargar la aplicación.

10.2.2.3 Cambiar idioma de la web

La página web está disponible en dos idiomas, español e inglés. Se puede cambiar el idioma de la página desde el selector de idioma, ilustración 10.29, situado en todas las páginas. En las páginas de identificación y descarga el selector se encuentra en la parte inferior derecha. En las páginas interiores el selector de idioma se sitúa en la parte superior derecha de la página.

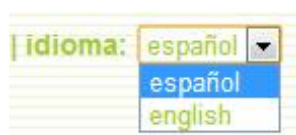


ILUSTRACIÓN 10.29: SELECTOR DE IDIOMAS

10.2.2.4 Consultar histórico

Para consultar el histórico de la monitorización de un dispositivo se ha de haber accedido correctamente a la aplicación, punto [10.2.2.1 Acceso a la aplicación](#). En la pantalla panel, ilustración 10.27, se puede acceder desde dos puntos a la página histórico. Desde el enlace situado en la parte superior derecha o desde los enlaces situados en las tablas de datos de los dispositivos. Desde cualquiera de las dos formas se accederá a la página histórico, ilustración 10.30.



ILUSTRACIÓN 10.30: MANUAL DE USUARIO: PÁGINA HISTORICO.

Para consultar el histórico se debe rellenar el formulario situado en la parte superior. En primer lugar se deberá seleccionar un usuario y un dispositivo en los selectores. Si se ha accedido a la página desde los enlaces situados en las tablas de datos de la página panel estos selectores ya estarán seleccionados. A continuación se deberá seleccionar una fecha en el calendario que se abrirá al pulsar sobre el cuadro de fecha. Después de la fecha se deberá introducir una hora. Esta fecha y hora marcarán el inicio del histórico, los dos siguientes campos de fecha y hora marcarán el final del histórico. Al tener todo seleccionado se debe pulsar el botón enviar.

10.2.2.5 Ver monitorización detallada

Para consultar el histórico de la monitorización de un dispositivo se ha de haber accedido correctamente a la aplicación, punto [10.2.2.1 Acceso a la aplicación](#). En la pantalla panel, ilustración 10.27, para acceder a la monitorización detallada de un dispositivo se ha de pulsar en el enlace “ver detalles” situado en la parte derecha de la barra donde está escrito el correo electrónico y el modelo del dispositivo que se quiere detallar. Al pulsar sobre el enlace la página se redireccionará a la página de detalles, ilustración 10.31.



ILUSTRACIÓN 10.31: MANUAL DE USUARIO: PÁGINA DE MONITORIZACIÓN DETALLADA

10.2.2.6 Resolución de problemas

En este punto se resolverán los principales problemas que puedan surgir en el uso de la aplicación.

10.2.2.6.1 No puedo acceder a la aplicación.

- Asegúrese de haber introducido correctamente los datos de solicitados, el correo electrónico y la contraseña que utilizó para registrarse en la aplicación móvil.

10.2.2.7 Preguntas más frecuentes

En este punto se darán respuestas a las preguntas más frecuentes que pueden surgir del uso de la aplicación.

10.2.2.7.1 ¿Cuáles son mis datos de acceso?

- Los datos de acceso son los introducidos al realizar el registro en la aplicación móvil.

10.2.2.7.2 ¿Puede acceder un usuario no registrado a la aplicación?

- No. Para acceder a la aplicación primero se ha tenido que registrar en la aplicación mediante la aplicación móvil.

10.2.2.7.3 ¿No me puedo registrar en la página web?

- El objetivo de esta aplicación es monitorizar los dispositivos móviles desde un servidor web remoto, por lo que no tiene sentido acceder a la aplicación web si no se tiene un terminal móvil monitorizado.

10.2.2.7.4 ¿Sólo puede descargarme la aplicación para Android?

- Sí, AndroidMonitor, como su nombre indica, solo es compatible con los dispositivos móviles que utilicen Android.

10.3 Anexo II: Contenido del disco

En este anexo se trazará el árbol de contenidos del disco que se adjunta con el presente documento. En el árbol figurarán los archivos más importantes del proyecto para facilitar su localización.

```

/doc/
Memoria.pdf
/src/
  app/
    assets/
    bin/
      classes/
      res/
        AndroidManifest.xml
        AndroidMonitor.apk
        classes.dex
        jarlist.cache
        resources.ap_
    gen/
    libs/
    res/
      drawable/
      drawable-hdpi/
      drawable-ldpi/
      drawable-mpdi/
      drawable-xhdpi/
      layout/
        activity_main.xml
        activity_monitor.xml
        activity_opciones.xml
        activity_perfil.xml
        login.xml
        signup.xml
      menu/
      values/
      values-large/
  src/
    es/
      uc3m/
        androidmonitor/
          MainActivity.java
          MenuPrincipalActivity.java
          Monitor.java
          MonitorService.java
          Opciones.java
          Perfil.java
        utils/
          DataBase.java
          Funciones.java
          iniciador.java
    .classpath
    .project
    AndroidManifest.xml
    Proguard-project
    Project.properties
bbdd/

```

```
androidmonitor.sql
web/
  aplicación/
    AndroidMonitor.apk
  css/
  img/
  js/
  lang/
  php_code/
    actualización.php
    cambiarPass.php
    cambiarPeriodo.php
    conexion.php
    consultarHistorico.php
    datosSensores.php
    historicoSensores.php
    moniSens.php
    monitorizarDispositivo.php
    panel.php
    registro.php
    selectDispositivo.php
  descargar.php
  footer.php
  histórico.php
  index.php
  monitorizarDispositivo.php
  panel.php
```

